Exploring Discriminative Speech Trajectory Structures

Heyun Huang, Louis ten Bosch, Bert Cranen, Lou Boves
Department of Linguistics, Radboud University Nijmegen, Erasmuslaan 1, 6525, Nijmegen, the Netherlands
{h.huang,l.tenbosch,b.cranen,l.boves}@let.ru.nl

Abstract
The articulators of the human speech production mechanism can only move relatively sluggishly. This results in speech sounds of which the acoustic speech properties mostly change continuously and gradually over time. However, such continuity constraints are seldom exploited for the purpose of discriminating different phones. In order to explore to what extent incorporating continuity information can help to improve phone discrimination, we investigated a multi-frame MFCC representation in combination with a supervised dimensionality reduction method which aims at finding a low-dimensional representation that best separates the different phones. The speech continuity information is encoded by a second-order smoothness regularizer. Experimental results on TIMIT phone classification show that the regularizer is helpful in better distinguishing vowels, but fails to improve the discrimination of consonants.

Index Terms: Dimensionality Reduction; Contextual Representation; TIMIT; regularization; Laplacian smoothing;

1. Introduction
Speech is generated by continuous movements of the articulators. Therefore, capturing the articulatory dynamics in parametric representations of the speech signal holds the promise of improving acoustic modeling in Automatic Speech Recognition (ASR) [1–5]. Specifically, capturing trajectories might eliminate the trajectory folding phenomenon in conventional ASR systems [3, 6].

Nearly all the aforementioned approaches tried to incorporate the trajectory information into generative acoustical models. However, generative models fail to utilize the information that discriminates between speech sounds. Therefore, it is worthwhile trying to develop representations that capture continuous trajectories in a discriminative framework. One way for doing this is by representing trajectories implicitly in the form of a number of consecutive frames in conjunction with supervised dimensionality reduction techniques such as Linear Discriminant Analysis (LDA) (e.g. [7–10]). However, conventional dimensionality reduction techniques do not automatically provide a good trade-off between the empirical risk (over-fitting of the high-dimensional data) and the structural risk (that the resulting representation does not reflect the actual degrees of freedom in the physical process that generated the (speech) data) [11]. Today, there are only few papers that address the structural risk issue in speech processing, e.g. [12, 13]. These papers have in common that they introduce regularization terms that might be implicitly related to the speech production process.

In this paper we investigate the application of a smoothness regularizer, which holds the promise of being able to find a good trade-off between the empirical and structural risk and of finding a low-dimensional representation that is discriminative and consistent with the physical production model [14]. In Section 2 we introduce this regularizer, and explain its relation to the speech production process. Section 3 briefly introduces several alternative approaches and explains the design of the experiments. Section 4 presents the results of the experiments. Discussions and conclusions are presented in Section 5.

2. The Smoothness Regularizer

2.1. Supervised Dimensionality Reduction
We develop the theory for a two-class problem: we have \( I_1 \) and \( I_2 \) tokens of two phones that we want to separate. All tokens are represented by a sequence of \( N \) MFCC frames (\( N = 23 \) in this paper); each frame comprises \( M = 13 \) coefficients. Thus, each token is represented as a matrix \( X_i \), \( i = 1, 2, \ldots, I_c, c \in [1, 2] \). Supervised dimensionality reduction algorithms first vectorize the matrix representation to \( x_i \in \mathbb{R}^D \), \( D = 13 \times 23 \), and then find the projection matrix \( W \in \mathbb{R}^{D \times d} \) with which a low-dimensional representations \( z_i \in \mathbb{R}^d \) can be obtained by \( z_i = W^T x_i \) that maximizes the separation between the two classes. \( W \) is found by

\[
\arg\max_W \left( \frac{\text{tr}(W^T S^{(b)} W)}{\text{tr}(W^T S^{(w)} W)} \right) \tag{1}
\]

In this paper we consider two different strategies for defining the within-class and between-class scatter matrices \( S^{(w)} \) and \( S^{(b)} \) in Eq. (1). In conventional Linear Discriminant Analysis (LDA) [15] all observations enter into the calculation; in Locally Discriminant Embedding (LDE) [8, 16], however, only a neighbourhood of ‘close’ observation pairs is taken into account. The size of the neighborhoods is determined by two LDE-parameters \( k_w, k_b \) (see [8]).

2.2. Enhancing Trajectory Continuity by Laplacian Smoothing
While it is difficult to estimate the exact number of degrees of freedom of the speech production system, it is evident that this number is much smaller than \( 23 \times 13 \), the number of dimensions that we have when we represent phones as a sequence of 23 frames of 13 MFCCs. If the ratio of the dimensionality of the raw data and the number of degrees of freedom in a physical system is very large, linear discriminant techniques are prone to over-fitting: spurious structure in the data that co-
incidentally distinguishes classes might be included in the discriminant function. This spurious structure will be visible in the form of high-frequency noise in the discriminant functions [17]. Low-dimensional underlying processes, on the other hand, will give rise to relatively smooth discriminant functions.

To retain the physical continuity in the discriminant function we need to impose continuity constraints on the projection matrix $W$. We can do this as follows. Consider the time trajectory of the $m$th MFC coefficient, $x_{m,n}$, $n = 1, 2, \ldots, N$, whose corresponding weights are represented by $w_m, w_{mn}, n = 1, 2, \ldots, N$. The degree of smoothness of the trajectory can be characterized by

$$J(w_m) = (w_{m1} - w_{m2})^2 + \sum_{i=2}^{n-2} (w_{mi} - 2w_{mi+1} + w_{mi+2})^2$$

$$+ (w_{m,n-1} - w_{mn})^2$$

To achieve a smooth $W$, the sum of the smoothness function over all MFCs ($J(W) = \sum_{m} J(w_m)$) must be minimized. Since we can write $J(W) = W^TDD^2W$, in which the $23 \times 23$ matrix $D$ is the Laplacian smoother (or Neumann discretizer) [17,18];

$$D = \frac{1}{23^2} \begin{pmatrix}
-1 & 1 & 0 & \cdots & 0 \\
1 & -2 & 1 & \cdots & 0 \\
1 & -2 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 1 & -2 & 1 & 0
\end{pmatrix}$$

After inserting the smoothness function into Eq. (1) and recombining factors, we obtain the smoothed supervised dimensionality reduction algorithm with the following criterion:

$$\text{argmax}_W \left( tr(W^T(S^{(a)}W) + \alpha S^{(b)} + \alpha DD^T) W^T \right)$$

(3)

The parameter $\alpha$ adjusts the balance between the smoothness function and $S^{(w)}$. With $\alpha = 0$ we obtain the original version of a dimensionality reduction algorithm specified by $S^{(w)}$ and $S^{(b)}$ in eq. (1).

3. Experimental Setup

3.1. The Data: TIMIT

We use TIMIT [19] for our experiments. The standard NIST training set, the core test set advised by TIMIT, and the development set by [20] are adopted for training, testing and parameter-tuning purposes. We adhere to the common practice and merge the original 64 phone labels defined by TIMIT into 48 phone labels; the glottal stops are not taken into account [21].

3.2. Classification Task

Our phone classification task is a multi-class classification problem. Here we decompose this multi-class classification task into multiple binary classification tasks [12,22]. Therefore, the overall multi-class classification performance depends crucially on that of the individual binary classifiers. Furthermore, the binary classification results are possibly more interpretable and informative for the further understanding of the acoustic representation of phones, which might be more instructive to the phonetic research, such as the acoustic modeling in ASR.

TIMIT has 44 classes of phones if all “silences” are excluded. Therefore, there are $44 \times (44 - 1)/2 = 946$ pairs of phones. In fact, many phone pairs are virtually never confused, for example vowels on the one hand and voiceless consonants on the other. Even some pairs from the same broad phonetic class are hardly ever confused. Because phone pairs with (almost) perfect separation may obscure the differences between approaches, we focused the comparison on those pairs whose classification accuracy numbers are lower than some thresholds: 0.95 and 0.90. Specifically, for each pair of phones, if any of the four compared approaches (that will be introduced in the next subsection) yields a classification accuracy lower than 0.90 or 0.95, this pair will be referred to as “most difficult” ($\leq 0.90$) and “less difficult” ($\leq 0.95$).

3.3. Compared Methods

We will compare four methods, namely LDA and LDE, and their smoothed variants SLDA and SLDE. The outputs of all these supervised dimensionality reduction methods are classified by a weighted $k$-Nearest-Neighbor (WkNN) classifier [7,8]. This WkNN works with a weighted majority voting: Suppose that $t_1, t_2, \ldots, t_k$ are the $k$ nearest neighbors of a test vector $t$. The weights of these neighbors are calculated by $w_i = \exp(-||t - t_i||^2/\tau)$, $i = 1, 2, \ldots, k$, and $t$ is assigned the class whose accumulated weights are largest.

With some of the four methods, there are some free parameters that need to be defined. By $r$ we denote the dimension after the dimensionality reduction step. LDA is parameter-free. LDE has two decisive parameters $k_w, k_b$, while SLDA and SLDE have one additional smoothness parameter $\alpha$. Finally, the WkNN classifier has two parameters $k$ and $\tau$, as explained above.

A grid search on the development data for the two WkNN parameters with $15 \leq k \leq 40$ and $3.5 \leq \tau \leq 6.5$ showed that their optimal setting did not depend critically on the dimension reduction method. Therefore, we select $k = 25$ and $\tau = 4.5$ for the configuration of the WkNN classifier. Any remaining free parameters are chosen by optimizing the performance on the development set using the following grid: $2 \leq k_w \leq 20, 2 \leq k_b \leq 40, \text{and} 0 \leq \alpha \leq 0.9, \text{with steps of} 2, 2, \text{and} 1. \text{The grid used for} r \text{is} 10 \leq r \leq 40 \text{with step size} 5$. Pilot experiments showed that a very low-dimensional representation ($r \leq 10$) yields very low accuracies.

4. Experimental Results

4.1. Classification Per Broad Class

Since different types of broad phonetic classes are likely to be affected differently by the production mechanism, this subsection shows the results for each broad phonetic class separately: Plosives (PL), Strong Fricatives (SF), Weak Fricatives (WF), Nasals (NS), Semi-Vowels (Se-V), Short Vowels (Sh-V), and Long-Vowels (Lo-V). As mentioned in Subsection 3.2, we always show the results of four compared methods on “crucial” binary pairs with two different thresholds: the most difficult pairs ($\text{acc} \leq 0.90$) and the less difficult pairs ($\text{acc} \leq 0.95$).

Firstly, Table 1 shows the average classification accuracy for vowel-like phones (Se-V, Sh-V, and Lo-V) and consonants (PL, SF, WF, and NS); the upper and lower pairs of data rows contain the results of the most difficult and less difficult tasks. This table shows that the vowel-like phones can benefit from the usage of smoothness regularization in both LDA and LDE cases. This seems to imply that the continuity information of
the most sonorant phones is more obvious and easier to capture. Unfortunately, using the trajectory information by smoothing hurts the performance of those consonant cases to some extent.

A further detailed analysis of the individual broad phonetic classes is given in Table 2. Here we decided to focus on the LDE/SLDE-based approaches, because they perform better than the conventional LDA and SLDA.

Table 1 shows the details underlying the data in Table 1. The deterioration in consonant classification is mostly due to plosives and nasals. The failure of using smoothness regularization is probably due to the event-like nature of these broad phonetic classes. Plosives are typically characterized by a discontinuous trajectory over time. However, the smoothness penalty probably forces the trajectory to be a smooth “curve”, which might obscure discriminative information from the discontinuity. Also for nasals the opening and closing of the velum may be expected to give rise to sudden jumps in the trajectories implying that the second-order smoother adopted in this paper might be counter-productive in characterising the trajectories associated with nasals. In other consonants, vowels and semi-vowels, SLDE tends to give slight improvements in classification accuracy compared to the non-smoothed version, suggesting that continuity information for these phones may indeed contribute to the discrimination between different speech sounds.

The two tables shown in this subsection clearly indicate that the trajectory information represented by the smoothness regularizer has a positive influence on the vowels and a negative impact on the consonants.

4.2. Analysis of α and r

Another approach to getting insight from the results is to analyze how the classification accuracy varies as a function of r (the dimension after dimensionality reduction) and the smoothness parameter α. To that end, we did the following analysis. For each pair of phones, we first select the 10-best combinations of (k, α) on the development set and then compute the average classification accuracy for each combination of (r, α). For vowels and consonants, these accuracies are shown in the form of the combined surface/contour plots of Fig. 1 and Fig. 2, respectively. Both figures show similar trends as observed in Table 1: the smoothness regularization has positive effects on the vowels, but not on the consonants. Another interesting observation concerns r: consonants have intrinsically a lower manifold dimension r than vowels.

To better visualize the impact of the smoothing operation, two similar plots are given for two types of vowels: long vowels (Fig 3) and short vowels (Fig 4). In the case of long vowels, an obvious increase along the “α” axis can be seen. Additionally, the maxima appear in the range where α approaches 1. This means that the broad phonetic class with the slowest-moving articulators, long vowels, does require smoothing of the trajectories by the second-order smoothness function. The other surface plot, for the short vowels, shows that the smoothness parameter is still important, but has a smaller impact than on long vowels. Apparently, the vowels with shorter durations are characterized

<table>
<thead>
<tr>
<th>Phone Class</th>
<th>FDA</th>
<th>SFDA</th>
<th>LDE</th>
<th>SLDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vowels (≤ 0.90)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consonants (≤ 0.90)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vowels (≤ 0.95)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consonants (≤ 0.95)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 1](image1.png)  
Classification accuracy of “less difficult” vowels as a function of reduced dimension r and smoothness parameter α.

![Figure 2](image2.png)  
Classification accuracy of “less difficult” consonants as a function of reduced dim r and smoothness parameter α.
In our future work we aim to analyze speech production mechanisms in order to build broad-class-specific or even phone-specific trajectory models for the purpose of improved discriminative training, rather than to use a single constraint (second-order smoother). Another aim is to investigate the robustness of the smoothness regularizer against certain noise types: the Neumann discretizer approach is theoretically expected to be robust to specific types of additive noise in the MFCC domain.

6. References