Sparse smoothing of articulatory features from Gaussian mixture model based acoustic-to-articulatory inversion: Benefit to speech recognition

Prasad Sudhakar¹, Prasanta Kumar Ghosh²

¹ICTEAM/ELEN, Universite catholique de Louvain, Belgium.
²Department of Electrical Engineering, Indian Institute of Science (IISc), Bangalore, India.

prasad.sudhakar@uclouvain.be, prasantg@ee.iisc.ernet.in

Abstract

Speech recognition using articulatory features estimated using Acoustic-to-Articulatory Inversion (AAI) is considered. A recently proposed sparse smoothing approach is used to postprocess the estimates from Gaussian Mixture Model (GMM) based AAI using Minimum Mean Squared Error (MMSE) criterion. It is well known that low-pass smoothing as post-processing improves the AAI performance. Sparse smoothing, on the other hand, not only improves the AAI performance but also preserves the MMSE optimality for as many estimates as possible. In this work we investigate the benefit of preserving MMSE optimality during postprocessing by using the smoothed articulatory estimates in a broad class phonetic recognition task. Experimental results show that the low-pass filter based smoothing results in a significant drop in the recognition accuracy compared to that using articulatory estimates without any smoothing. However, the recognition accuracy obtained by articulatory features from sparse smoothing is similar to that using articulatory features directly from GMM based AAI without any postprocessing. Thus, sparse smoothing provides benefit both in terms of the inversion performance as well as recognition accuracy, while that is not the case with low-pass smoothing.

Index Terms: phonetic recognition, acoustic-to-articulatory inversion, smoothing, Gaussian mixture model, sparsity, Chambolle-Pock, ℓ1 minimization

1. Introduction

The task of estimating articulatory representation from acoustic features is referred to as the acoustic-to-articulatory inversion (AAI). The choices of acoustic and articulatory representations generally depend on the applications. One motivation behind ongoing research on AAI comes from its potential applications such as automatic speech recognition where the estimated articulatory features could be incorporated. It has been shown that articulatory information could improve the recognition performance [1, 2, 3, 4]. Since direct articulatory measurements are not readily available for everyday applications, AAI plays a crucial role in serving as an ancillary method. Apart from speech recognition, articulatory information estimated from inversion could be potentially useful for speech synthesis [5, 6], speaker recognition [7], computer assisted language learning, as automatic articulatory feedback could be given on the student’s pronunciation [8, 9].

The acoustic-articulatory map can be learnt in a number of ways – statistical models such as Gaussian mixture model (GMM) [10], mixture density network (MDN) [11], trajectory hidden Markov model (HMM) [12] and generalized smoothness criterion (GSC) [13], episodic memory based approach [14], codebook approach [15] and Neural Network approaches [16]; a comprehensive summary of different mapping techniques can be found in [17]. It has been shown that the AAI performance improves by smoothing, either as a post-processing [10] step or directly integrated into the inversion criterion [10, 13, 18]; this is mainly because the realistic articulatory trajectories are smooth and slowly varying [13]. In our work, we use GMM for modeling the statistical mapping [10], where the articulatory features are estimated using the minimum mean squared error (MMSE) criterion. We consider smoothing in the post-processing step.

Smoothing in post-processing, typically done by low-pass filtering, makes the articulatory trajectories smooth and realistic. However, it is important to note that the samples of the smoothed trajectories no longer remain the same as the ones obtained from GMM based AAI, and hence, are no longer optimal in the MMSE sense. However, maintaining the MMSE optimality of the samples could be crucial when the smoothed articulatory features are to be used for speech recognition [19, 20, 21, 2]. The recently proposed sparse smoothing [22] is designed to preserve the MMSE optimality of as many samples as possible and to improve the AAI performance. Sparse smoothing approach advocates smoothing of the estimated articulatory trajectories by adding a sparse correction sequence as opposed to a convolutional based smoothing in low-pass filtering.

The major advantage of sparse smoothing is its ability to balance the degree of smoothness and the number of samples for which MMSE optimality is preserved. However, its potential benefit in speech recognition using smoothed articulatory features remains to be investigated. Thus, the goal in this work is to compare the recognition performance using articulatory features, smoothed by low-pass filtering and sparse smoothing. Note that the goal here is not to achieve the best recognition performance using articulatory features rather to examine the effect of smoothing on recognition. The differences in the working principles of two types of smoothing could affect the recognition in different ways. The study in this paper analyzes the interplay between AAI performance and recognition accuracy by using two smoothing schemes. Broad class phonetic recognition experiments with articulatory dataset reveal that the recognition performance drops by 7.8% (absolute) when using the low-pass filtering compared to when no smoothing is performed, while there is no significant drop in the recognition accuracy when the trajectories are sparsely smoothed. This result suggests that preserving MMSE optimality is crucial to obtain better recognition accuracy. We begin by briefly describing the sparse smoothing approach.
2. Sparse smoothing

Sparse smoothing problem is solved by posing it as an optimization problem [22]. As the goal is to simultaneously smooth and maintain the sparsity of the correction, the objective function contains a term that quantifies the smoothness and a term that quantifies the sparsity of the correction. The smoothness of the articulatory trajectory is quantified by the energy of its high frequency content: lower the energy in the high frequencies, smoother the trajectory is, and vice-versa. The sparsity of the correction term is quantified using its \( \ell_1 \) norm\(^1\).

2.1. Formulation of the optimization problem

Let \( w_n \) denote the acoustic feature vector at the \( n \)-th frame and the corresponding articulatory feature vector be denoted by \( x_n = [x_1, \ldots, x_J]^T \), where \( x_j \) is the \( j \)-th articulatory feature and there are a total of \( J \) articulatory features; \( T \) is the transpose operator. In AAI, the articulatory features are estimated from the given acoustic feature sequence \( w_n \), \( 1 \leq n \leq N \) of a sentence of length \( N \) frames. \( x_n \) and \( w_n \) are statistically mapped through GMM and the estimated articulatory feature vector \( \hat{x}_n \) is obtained using minimum mean squared error (MMSE) criterion [10].

Let \( \hat{x}_{n,j} \) denote the smoothed \( j \)-th articulatory feature trajectory obtained using zero-phase filtering through convolution with a low pass filter \( h_j \), with cut-off frequency \( f_j \) specific to the \( j \)-th articulator. In general, the samples of \( \hat{x}_{n,j} \) differ from the optimal estimate \( \hat{x}_{0,j} \). However, sparse smoothing produces a trajectory \( \hat{x}_{n,j}^{ss} \) that differs from \( \hat{x}_{n,j} \) at only few locations. Thus the difference

\[
d_{n,j} = \hat{x}_{n,j}^{ss} - \hat{x}_{n,j} \quad (1)
\]

is sparse and hence several MMSE optimal samples are retained. To measure the smoothness of the articulatory feature vector, a suitable high-pass filter \( g_j \) is chosen with cut-off frequency \( f_j \) specific to \( j \)-th articulator and the term \( \|g_j \ast (\hat{x}_{n,j} + d_{n,j})\|_2 \) is minimized. Simultaneously, the sparsity of \( d_{n,j}' \) is enforced by minimizing its \( \ell_1 \) norm.

Denoting \( \tilde{x} \) as \([\tilde{x}_1, \ldots, \tilde{x}_N]^T \) and \( d = [d_1, \ldots, d_N]^T \) respectively, and observing that zero-phase filtering with a \( N \)-length filter \( g_j \) is equivalent to a multiplication by the convolution matrix \( G^j \), constructed using the autocorrelation sequence of \( g_j \); \( G^j = \sum_n g_{-n}g_{n-j} \), sparse smoothing is achieved by solving the following optimization problem

\[
d_{n,j}' = \arg \min_{d} \|d\|_1 \text{ subject to } \|G^j(d') - y_j\|_2 \leq \epsilon, \quad (2)
\]

where \( y_j = -G^j \tilde{x} \) and \( \epsilon \) is the amount of tolerance on the smoothness, and setting \( \tilde{x}^{ss} = \tilde{x} + d_{n,j}' \). The problem in (2) is in the standard Basis Pursuit DeNoising (BPDN) [23] form and related to LASSO [24].

2.2. Numerical method

We use a primal-dual method called the Chambolle-Pock (CP) algorithm [25] to solve the optimization problem in Eq. (2). Though a plenty of toolboxes such as SPGL1, CVX, etc. are available for solving \( \ell_1 \) minimization problems, the CP algorithm has the unique advantage of being generic, flexible, and also having convergence guarantees under broad conditions. Further, CP algorithm relies on the proximal operators of functions, which are easy to evaluate. We shall briefly describe the algorithm in this section for the sake of completeness.

Let \( K : \mathbb{R}^N \rightarrow \mathbb{R}^M \) be a continuous linear operator with norm \( \|K\| < \infty \). Let \( F : \mathbb{R}^M \rightarrow [0, +\infty] \) and \( G : \mathbb{R}^N \rightarrow [0, +\infty] \) be two proper, convex, lower-semicontinuous functions. The CP algorithm in Algorithm 1, is used to solve saddle-point problems obtained from the primal problem

\[
\min_{u \in \mathbb{R}^N} F(Ku) + G(u). \quad (3)
\]

**Algorithm 1: Chambolle-Pock**

- **Input:** Choose \( \tau, \sigma > 0 \), \( (u^0, v^0) \in \mathbb{R}^N \times \mathbb{R}^M \), \( \theta \in [0, 1] \) and \( u^0 = u^0 \).
- **Iterate:** For \( n \geq 0 \), until stopping criterion

\[
u^{n+1} = \text{prox}_{\sigma F}(u^n + \sigma K \hat{u}^n)
\]

\[
u^{n+1} = \text{prox}_{\tau G}(u^n - \tau K^* v^{n+1})
\]

\[
u^{n+1} = u^{n+1} + \theta(u^{n+1} - u^n)
\]

- **Output:** \((u^n, v^n)\)

\( F^* \) is the convex conjugate of \( F \) and \( K^* \) is the adjoint of \( K \). The explicit forms of the proximal operators depend on the functions \( F \) and \( G \). The general theory about proximal operators and algorithms can be found in [26]. To ensure the convergence of the algorithm, the parameters \( \tau \) and \( \sigma \) have to be chosen such that \( \tau \sigma \|K\|^2 < 1 \).

In order to express the sparse smoothing problem Eq. (2) in the unconstrained form Eq. (3), we consider the indicator functions of the convex sets defined by the constraints. The convex indicator function \( \iota_C(\cdot) \) on a convex set \( C \) is defined by

\[
\iota_C(z) = \begin{cases} 0 & \text{if } z \in C, \\ +\infty & \text{otherwise.} \end{cases}
\]

Consider the convex norm ball

\[
B^\epsilon = \{ z \in \mathbb{R}^N \mid \|z - y\|_2 \leq \epsilon \}.
\]

By including the convex indicator function of the set \( B^\epsilon \) into the objective function, the constrained problem is turned into the following unconstrained problem, which can be solved using CP algorithm:

\[
d_{n,j}' = \arg \min_{d \in \mathbb{R}^N} \left( \|d\|_1 + \iota_{B^\epsilon} (G^j(d')) \right).
\]

The sparsely smoothed \( j \)-th articulatory feature trajectory is then given by \( x_{n,j}^{ss} = \tilde{x}_{n,j} + d_{n,j}' \). The proximal operator for the \( \ell_1 \)-norm is the simple component-wise soft-thresholding operator, defined for a scalar \( u \) as:

\[
\text{prox}_{\gamma \ell_1}(u) := \begin{cases} 0 & \text{if } |u| \leq \gamma, \\ (|u| - \gamma) \text{sgn}(u) & \text{otherwise}. \end{cases}
\]

The proximal operator of the function \( \iota_{B^\epsilon}(u) \) is the following projection function onto convex set \( B^\epsilon \):

\[
\text{prox}_{\iota_{B^\epsilon}}(u) := y^\epsilon + (u - y^\epsilon) \min \{1, \epsilon/(\|u - y^\epsilon\|_2)\}.
\]

The proximal operator of the conjugate function \( F^* \) is easily computed using the celebrated Moreau’s identity [25]. In effect, Algorithm 1 simply involves repeatedly evaluating simple functions and hence has minimal requirements on computational resources.
3. Dataset and pre-processing
For all the experiments in this paper, we use the Multichannel Articulatory (MOCHA) database [27] that contains speech and corresponding ElectroMagnetic Articulography (EMA) data from one male and one female speaker of British English. The EMA data consist of dynamic positions of the EMA sensors in the midsagittal plane of the talker. Seven sensors are placed on upper lip (UL), lower lip (LL), lower incisor (LI), tongue tip (TT), tongue body (TB), tongue dorsum (TD), and velum (VEL)). Thus, we use 14 dimensional raw EMA features for representing articulatory space (i.e., X and Y co-ordinates of seven EMA sensors), namely ULx, LLx, Llx, TTx, TBx, TDx, VELx, ULy, LLy, LLy, TTLy, TBy, TDLy, VELy. Following the preprocessing steps outlined in [13], we obtain parallel acoustic and articulatory data at a frame rate of 100 observations per second. Acoustic feature MFCCs (39 dimensional, including first and second derivatives) are computed using 20 msec frame length with 10 msec shift [28].

4. Experiment and results
4.1. Experimental setup
The AAI as well as broad class phonetic recognition experiments are performed separately on the male and female subjects of the MOCHA database in a ten fold cross validation setup. Parallel acoustic and articulatory data of nine folds are used as the training set for estimating the GMM parameters which models the acoustic-articulatory map. Remaining one fold is used as the test set for which the articulatory features are estimated given the test utterance’s acoustic. Following the work by Toda et al. [10], we have used 64 mixture GMM for learning the acoustic-articulatory map using the training data.

The smoothing filter parameters to be used in the post-processing step are optimized on the training set itself. To ensure that the filter parameters are optimized on a set of sentences which are not used for training the GMM, we perform an AAI using nine-fold cross-validation setup within the training set, i.e., every eight folds are used to train a GMM to model acoustic-articulatory map which is used on the remaining fold of the training set for estimating articulatory features. Thus we obtain estimates of the articulatory features for each training utterance. These estimated articulators of the training utterances along with original articulatory trajectories are used to optimize the hyper-parameters of the sparse smoothing optimization, i.e., \( f_0, \sigma, \) and \( \epsilon \). The set of values \( \{3, 4, 5, \ldots, 24\} \) Hz are chosen for \( f_0 \), \( \{e^{-5}, e^{-4}, e^{-3}, 0.01, 0.1\} \) for \( \sigma \) and \( \{0.01, 0.1, 1, 10, 50, 100, 500, 1000\} \) for \( \epsilon \). The combination of hyper-parameters which yields the best performance is finally chosen. Articulator specific high-pass filter \( g_0 \) is chosen as a 5-order Chebyshev type-II IIR filter (using cheby2 () in Matlab) with stop-band ripple 40 dB down compared to the pass-band ripple [29]. We also use a typical low-pass filter based smoothing for post-processing to compare with the performance of the sparse smoothing approach. Articulator specific low-pass filters are chosen to be 5-order Chebyshev type-II IIR filter with stop-band ripple 40 dB down compared to the pass-band ripple. The cut-off frequencies of these low-pass filters are also optimized on the estimated articulatory features of the training set. The cut-off frequencies for this optimization are chosen from 3Hz to 24Hz with a step size of 1Hz.

Once the parameters of sparse smoothing as well as low-pass filters are optimized, they are used to postprocess articulatory estimates of all the utterances of the training set as well as the test set. Henceforth, we refer to the GMM based inversion followed by sparse smoothing and low-pass smoothing as GMM_{SS} and GMM_{NS} respectively. We perform a broad class phonetic recognition experiment using smoothed articulatory estimates in a ten fold cross-validation setup identical to the inversion experiments, i.e., the smoothed articulatory estimates of the training utterances from nine folds are used for training the recognizer and those from the remaining one fold are used as the test set for recognition. We use four broad phonetic categories for our experiments following the broad phonetic categories used by Sainath et al [30] in their phonetic recognition task. The broad phonetic classes are vowels, fricatives, stops and nasals. The phone boundaries are obtained by performing a forced alignment on each sentence using the corresponding transcription (with a base set of 39 English phonemes). Each subject in the MOCHA-TIMIT corpus has recordings of 460 sentences (~ 20 min.). It should be noted that due to the limited data size, the number of frames for each of the 39 phonemes is limited, causing poor training of the recognizer. This is one of the reasons for grouping them to just four broad phonetic categories.

The recognition experiments are performed using hidden Markov model toolkit (HTK) [28]. Three state left-to-right HMMs are used for modeling each of the broad class phonemes. The number of components in GMM for emission probability density of each state is varied from 1 to 256. Results are reported using 256 number of components since it yields the best recognition accuracy. Recognition is performed using the smoothed articulatory trajectories and their first and second derivatives (i.e., 42 dimensional articulatory features). In addition to the recognition accuracies using GMM_{SS} and GMM_{NS}, we also report recognition accuracy using articulators directly from the GMM based inversion without any smoothing in the post-processing step (referred to as GMM_{NS}). Recognition using original articulators are also reported; these act as the upper bound on the recognition accuracies that could be obtained by the articulatory features. For comparison, the recognition accuracies using MFCCs are reported too. Note that in [22], the inversion experiments were performed in a 5-fold setup on the MOCHA corpus unlike 10-fold in the present work. Hence, we also compare the inversion performance using GMM_{SS}, GMM_{S}, and GMM_{NS}. The inversion performances are measured using the RMSE and Pearson Correlation Coefficient (PCC) between the original and smoothed articulatory trajectories, as done in [22].

4.2. Results and discussions
Table 1 shows the broad class phonetic recognition accuracies using articulatory features estimated by GMM_{NS}, GMM_{SS} as well as original articulatory (denoted by Orig) and acoustic features (MFCC). The recognition accuracies are reported for each of the ten folds separately in case of male and female subjects used in the experiment. It is clear that the recognition accuracy using GMM_{SS} is lower than that using GMM_{NS} scheme for each fold. In fact, the recognition accuracy over all folds using GMM_{S} is significantly (\( p < 0.01 \)) less (6.61% and 8.98% absolute drop in recognition for male and female subjects respectively) than that using GMM_{NS} scheme. This could be because, low pass filtering in GMM_{S} are optimized for the RMSE between original and smoothed articulators. RMSE is a global measure and it may not preserve the discrimination among broad phonetic classes’ articulatory estimates obtained through MMSE criterion. Better recognition accuracy using articulatory features from GMM_{NS} over GMM_{S} also suggests

\[2\text{It should be noted that due to limited data, we have merged strong and weak fricatives into one class called 'fricative' unlike that in [30]; the stop closures are also included in the 'stop's' category.}\]
that the MMSE estimate using GMM based AAI (without any postprocessing) provides considerable information for discriminating different broad phonetic classes and smoothing (postprocessing) can reduce the discrimination if it is done only to maximize AAI performance. This is because improvement in AAI performance may not necessarily imply better discrimination among broad phonetic classes. Hence, if the target application following AAI is speech recognition, it would be useful to preserve the MMSE estimates as much as possible, which is, in fact, the principle of sparse smoothing.

![Figure 1: Comparison of AAI performance of low-pass filtering (smoothing) and sparse smoothing using RMSE.](image)

![Figure 2: Comparison of AAI performance of low-pass filtering (smoothing) and sparse smoothing using PCC.](image)

Table 1: Broad class phonetic recognition accuracies (in percent) using articulatory estimates with and without smoothing. Recognition accuracies using original articulators and MFCCs are also reported for comparison.

<table>
<thead>
<tr>
<th>Fold Number</th>
<th>Male Subject</th>
<th>Female Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GMM_NS</td>
<td>GMM_S</td>
</tr>
<tr>
<td>1</td>
<td>64.99</td>
<td>55.08</td>
</tr>
<tr>
<td>2</td>
<td>65.42</td>
<td>59.74</td>
</tr>
<tr>
<td>3</td>
<td>65.67</td>
<td>59.14</td>
</tr>
<tr>
<td>4</td>
<td>64.11</td>
<td>57.00</td>
</tr>
<tr>
<td>5</td>
<td>65.08</td>
<td>57.88</td>
</tr>
<tr>
<td>6</td>
<td>66.55</td>
<td>61.66</td>
</tr>
<tr>
<td>7</td>
<td>65.74</td>
<td>58.79</td>
</tr>
<tr>
<td>8</td>
<td>66.13</td>
<td>59.09</td>
</tr>
<tr>
<td>9</td>
<td>63.68</td>
<td>59.01</td>
</tr>
<tr>
<td>10</td>
<td>67.54</td>
<td>61.21</td>
</tr>
<tr>
<td>Average</td>
<td>(65.47±1.06)</td>
<td>(58.86±1.59)</td>
</tr>
</tbody>
</table>

We study the effect of smoothing as post-processing in the GMM based AAI on the recognition accuracy using smoothed articulatory features. It is found that articulatory features smoothed using sparse smoothing results in better accuracy in broad class phonetic recognition task than those using low-pass filter based smoothing although both of them have similar AAI performance. It is also found that the recognition accuracy using sparse smoothing is similar to that using no smoothing. This suggests that preserving the optimality of the AAI criterion during smoothing (as done in sparse smoothing) is crucial to preserve the recognition accuracy of no smoothing. Thus, the sparse smoothing offers benefit in terms of both the AAI performance and the recognition accuracy unlike the low-pass filter based smoothing. While only broad phonetic classes are considered in this work, the recognition benefit needs to be investigated with larger set of phonetic units from larger database and using subject-independent AAI. The recognition benefit of sparsely smoothed articulatory features also needs to be investigated when they are used jointly with acoustic features. These are parts of our future work.

5. Conclusions
6. References


