On the Acoustic-to-Electropalatographic Mapping

Asterios Toutios and Konstantinos Margaritis

Parallel and Distributed Processing Laboratory, Department of Applied Informatics,
University of Macedonia, Thessaloniki, Greece
{toutios, kmarg}@um.ac.gr

Abstract. Electropalatography is a well established technique for recording information on the patterns of contact between the tongue and the hard palate during speech. It leads to a stream of binary vectors, called electropalatograms. We are interested in the mapping from the acoustic signal to electropalatographic information. We present results on experiments using Support Vector Classification and a combination of Principal Component Analysis and Support Vector Regression.

1 Introduction

Electropalatography (EPG) [1] is a widely used technique for recording and analyzing one aspect of tongue activity, namely its contact with the hard palate during continuous speech. It is well established as a relatively non-invasive, conceptually simple and easy-to-use tool for the investigation of lingual activity in both normal and pathological speech. An essential component of EPG is a custom-made artificial palate, which is moulded to fit as unobtrusively as possible against a speaker’s hard palate. Embedded in it are a number of electrodes (62 in the Reading EPG system, which is considered herein). When contact occurs between the tongue surface and any of the electrodes a signal is conducted to an external processing unit and recorded. Typically, the sampling rate of such a system is 100-200 Hz. Thus, for a given utterance, the sequence of raw EPG data consists of a stream of binary (1 if there is a contact; 0 if there is not) vectors with both spatial and temporal structure. Figure 1 shows part of such a stream. Observation of both temporal and spatial details of contact across the entire palatal region can be very helpful to identify many phonetically relevant details of lingual activity.

Electropalatography has been successfully used to study a number of phenomena in phonetic descriptive work, in studies of lingual coarticulation and in the diagnosis and treatment of a variety of speech disorders. It has also been suggested that visual feedback from EPG might also be used in the context of second language.

However, there are difficulties in acquiring EPG data. First, each artificial palate must be individually manufactured from dental moulds of the speaker. Second, the artificial palate in the speaker’s mouth may sometimes hinder their ability to produce normal speech.
What is suggested here is that some means of estimating EPG information using only the audio signal (which is far more easier to record and handle) as a source would be beneficial. To this end, we study the mapping from the acoustic signal to the EPG vectors, namely the *acoustic-to-electropalatographic mapping*. We adopt a machine learning point of view, in the sense that we try to infer the mapping only *from the data*, without making a priori use of any kind of speech production related theoretical intuitions.

2 The MOCHA Database

The MOCHA (Multi-Channel Articulatory) [2] database is evolving in a purpose built studio at the Edinburgh Speech Production Facility at Queen Margaret University College.

During speech, four data streams are recorded concurrently straight to a computer: the acoustic waveform, sampled at 16kHz with 16 bit precision, together with laryngograph, electropalatograph and electromagnetic articulograph data. EPG provides tongue-palate contact data at 62 normalised positions on the hard palate, defined by landmarks on maxilla. The EPG data are recorded at 200Hz.

The speakers are recorded reading a set of 460 British TIMIT sentences. These short sentences are designed to provide phonetically diverse material and capture with good coverage the connected speech processes in English. All waveforms are labelled at the phonemic level.

The final release of the MOCHA database will feature up to 40 speakers with a variety of regional accents. At the time of writing this paper two speakers are available. For the experiments herein, the acoustic waveform and EPG data, as well as the phonemic labels for the fsew0 speaker, a female speaker with a Southern English accent, are used.
3 Overview of Machine Learning Techniques used

3.1 C-Support Vector Classification

Given $n$ training vectors $x_i$ in two classes and a vector $y \in \mathbb{R}^n$ such that $y_i \in \{-1, 1\}$, we want to find a decision function that separates the two classes in an optimal (from a Structural Risk Minimization perspective) way [3–5]. The decision function that the C-SVC algorithm gives is:

$$f(x) = \text{sgn}(\sum_{i=1}^{n} a_i y_i k(x, x_i) + b),$$

(1)

where the $a$ coefficients are the solution of the quadratic programming problem:

$$\text{maximize } W(a) = -\frac{1}{2} \sum_{ij} a_i a_j y_i y_j k(x_i, x_j)$$

subject to $0 \leq a_i \leq C, i = 1, \ldots, n$, and $\sum_i a_i y_i = 0.$

(2)

Here $C$, called the penalty parameter, is a parameter defined by the user and $k(x_i, x_j)$ is a special function called the kernel which serves to convert the data into a higher-dimensional space in order to account for non-linearities in the decision function. A commonly used kernel is the Radial Basis Function (RBF) kernel:

$$k(x, y) = \exp(-\gamma \| x - y \|^2),$$

(3)

where the $\gamma$ parameter is selected by the user.

3.2 $\epsilon$-Support Vector Regression

The $\epsilon$-SVR algorithm [6,5] generalizes the C-SVC algorithm to the regression case. Given $n$ training vectors $x_i$ and a vector $y \in \mathbb{R}^n$ such that $y_i \in \mathbb{R}$, we want to find an estimate for the function $y = f(x)$. According to $\epsilon$-SVR, this estimate is:

$$f(x) = \sum_{i=1}^{n} (a_i^* - a_i) k(x_i, x) + b,$$

(4)

where the coefficients $a_i$ and $a_i^*$ are the solution of the quadratic problem

$$\text{maximize } W(a, a^*) = -\epsilon \sum_{i=1}^{n} (a_i^* + a_i) + \sum_{i=1}^{n} (a_i^* - a_i) y_i - \frac{1}{2} \sum_{i, j=1}^{n} (a_i^* - a_i)(a_j^* - a_j) k(x_i, x_j)$$

subject to $0 \leq a_i, a_i^* \leq C, i = 1, \ldots, n$, and $\sum_{i=1}^{n} (a_i^* - a_i) = 0.$

(5)

$C > 0$ and $\epsilon \geq 0$ are chosen by the user.
3.3 Principal Component Analysis

PCA [7,1] is a transform that chooses a new coordinate system for a data set such that the greatest variance by any projection of the data set comes to lie on the first axis, the second greatest variance on the second axis, and so on. The new axes are called the principal components. PCA is commonly used for reducing dimensionality in a data set while retaining those characteristics of the dataset that contribute most to its variance by eliminating the later principal components.

The direction \( \mathbf{w}_1 \) of the first principal component is defined by

\[
\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|_2 = 1} E\{\mathbf{w}^T \mathbf{x}^2\}
\]

where \( \mathbf{w}_1 \) is of the same dimension as the data vectors \( \mathbf{x} \). Having determined the direction of the first \( k-1 \) principal components, the direction of the \( k \)th component is:

\[
\mathbf{w}_k = \arg \max_{\|\mathbf{w}\|_2 = 1} E\{\mathbf{w}^T (\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i^T \mathbf{x})^2\}.
\]

In practice, the computation of the \( \mathbf{w}_i \) can be simply accomplished using the sample covariance matrix \( E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{C} \). The \( \mathbf{w}_i \) are then the eigenvectors of \( \mathbf{C} \) that correspond to the largest eigenvalues of \( \mathbf{C} \).

4 Data Processing

The MOCHA database includes 420 utterances of the fsfew0 speaker. For ease of experimentation, we use a small subset of 35 utterances that span across the whole set. We process these data as follows.

First, based on the label files we omit silent parts from the beginning and end of the utterances. During silent stretches the tongue can possibly take any configuration, something that could pose serious difficulties to our task.

Next, we perform a standard Mel Frequency Spectral Analysis [8] on the acoustic signal with the VOICEBOX Toolkit [9], using a window of 16ms (256 points) with a shift of 5ms (this is to match the 200Hz sampling rate of the EPG data). We use 30 filterbanks and calculate the first 13 Mel Frequency Cepstral Coefficients. Then, we normalize them in order have zero mean and unity standard deviation.

In order to account for the dynamic properties of the speech signal and cope with the temporal extent of our problem, we just use a commonplace in the speech processing field \textit{spatial metaphor for time}. That is, we construct input vectors spanning over a large number of acoustic frames. Based on some previous small-scale experiments of ours, we construct input vectors consisting of the MFCCs of 15 frames: the frame in question, plus the 7 previous ones, plus the 7 next ones.
Thus, we end up with training examples with a 195-dimensional (15 x 13) real-valued vector as input and a 62-dimensional binary vector as output. Using 28 utterances for the training set and 7 for the test set, we have 14408 training examples and 3321 test examples.

5 Experiments

5.1 Assuming Independencies - SVC

As a first approach to finding a mapping from the MFCCs to the EPG data, we make the working assumption that every EPG event (a contact or a non-contact at a certain electrode and point in time) is independent of neighbouring (in space and time) EPG events. Thus, the problem of estimating EPG patterns, becomes a problem of training 62 binary classifiers.

Considering each EPG point as a separate case we find that our output data are quite unbalanced, i.e. there are big differences in the number of positive (contacts) and negative (non-contacts) examples (Figure 2 is a schematic of the distributions in the test set - the distributions in the training set are similar). In order to take this fact into account we use a weighted variance of the C-SVC algorithm. The quadratic problem 8 is slightly changed into:

\[
\text{maximize } W(\mathbf{a}) = -\frac{1}{2} \sum_{ij} a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\
\text{subject to } 0 \leq a_i \leq C^+ \text{ if } y_i = +1, \ 0 \leq a_i \leq C^- \text{ if } y_i = -1 \text{ and } \sum_i a_i y_i = 0.
\]

That is, we consider different penalty parameters for the two classes. A usual heuristic is that \(C^+/C^- = l^-/l^+ = \alpha\) where \(l^+\) and \(l^-\) are the number of positive and negative examples respectively. For our experiment, we use \(C^- = (1 + \alpha)/2\alpha\)

\[\text{Fig. 2. Distributions of EPG events (in the test set). The bigger the square, the bigger the difference between positive and negative examples. Black squares indicate excess of positive examples and white squares excess of negative examples.}\]
and \( C^+ = (1 + \alpha)/2 \). We consider the RBF kernel with \( \gamma = 1 \). The experiments are conducted with the LIBSVM software package [10].

Our results are shown in Figure 3 and Table 1.

**Fig. 3.** Results of C-SVC experiment. (a) Overall Classification Rate (b) Classification Rate of Contacts. (c) Classification Rate of Non-Contacts. The largest the black square the closest the rate to 100%. White squares indicate that there are no examples of that class in the test set.

### 5.2 Accounting for Spatial Relationships - PCA and SVR

In order to account for the spatial relationships in the EPG data we consider applying PCA. As a first step, we perform PCA for various numbers of principal components and measure the relative reconstruction errors as:

\[
\text{reconstruction error} = \frac{1}{62 \times n} \sum_{i=1}^{n} \sum_{j=1}^{62} |y_{orig, i, j} - y_{rec, i, j}|
\]

where \( n \) is the number of output vectors, \( y_{orig} \) are the original EPG events (1 for a contact, 0 for a non-contact) and \( y_{rec} \) are the reconstructed EPG events after applying PCA, inverting PCA back to the original input space and forcing the output values to be either 1 or 0. The reconstruction results are shown in Figure 4.

The idea here is to select a number of (directions of) principal components, perform SVR on the components themselves and then revert back to the original output space. We select the first 30 principal components which give reconstruction error 0.64% on the test set. Figure 5 shows the first 12 principal components.

For the SVR part of the experiment we use the \( \varepsilon \)-SVR algorithm of the LIBSVM software. We use \( C = 1 \) and the RBF kernel with \( \gamma = 0.0026 \). The results, after the PCA inversion, are shown in Figure 6 and Table 1.

### 6 Conclusion

We applied two methods to the acoustic-to-electropalatographic mapping task, the first of which does not take into account the spatial dimension inherent in the
**Fig. 4.** PCA reconstruction results (as percentages).

**Fig. 5.** Principal Components of EPG data. The largest the square the bigger the absolute value. Black squares indicate positive values and white squares negative values.

**Fig. 6.** Results of PCA+SVR experiment. (a) Overall Classification Rate (b) Classification Rate of Contacts. (c) Classification Rate of Non-Contacts. The largest the black square the closest the rate to 100%. White squares indicate that there are no examples of that class in the test set.

<table>
<thead>
<tr>
<th></th>
<th>Chance Level</th>
<th>Weighted SVM</th>
<th>PCA+SVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>55.9977</td>
<td>88.8432</td>
<td>91.8189</td>
</tr>
</tbody>
</table>

**Table 1.** Chance Level of the test set and Overall Classification Rates of the methods used.
EPG data, while the second one does. The classification rates we achieved exceed (though not by far) the chance level of the data, defined as the average percentage of the class with the most examples among the EPG points. The PCA+SVR method gives better overall results, though there are not big differences at the distributions of the errors between the two methods. We believe that there is room for improvement, especially at the PCA+SVR case.

One thing is that we should experiment with more data. The amount of data in the MOCHA database offers such an opportunity. Training time is always an issue, but recent findings in the machine learning field, such as Cross-Training [11], seem quite promising in the direction of speeding up things.

Finally, the temporal dimension of the problem might also be taken into account. There are promising proposals in that area as well, such as the HMM-SVM method [12].

References