LEARNING ELECTROPALATOGRAMS FROM ACOUSTICS

Asterios Toutios and Konstantinos Margaritis

Parallel and Distributed Processing Laboratory, Department of Applied Informatics
University of Macedonia, Thessaloniki, Greece
{toutios,kmarg}@uom.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, leading to a stream of binary vectors called electropalatograms, consisting of elecropalatographic events – contacts or non-contacts between the tongue and the palate. A data-driven approach to mapping the speech signal onto electropalatographic information is presented. A combination of Principal Component Analysis and Support Vector Regression is used, yielding classification scores of more than 93% on individual electropalatographic events, for a single speaker. This may be viewed as a special case of the, well-known in the speech community, speech inversion problem which refers to inferring production parameters from the speech signal.

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; -1 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 succesfully 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 (an exhaustive listing of related publicatons can be found in [2]). It has also been suggested that visual feedback from EPG might be used in the context of second language acquisition.

We are exploring the mapping from the acoustic speech signal to electropalatographic information. In other words, we want to train a system that, when presented with previously unobserved acoustics, outputs the corresponding EPG sequences. We adopt an entirely data-driven approach, i.e. we do not use any speech production intuitions at all. We use a combination of Principal Component Analysis and Support Vector Regression for the task.

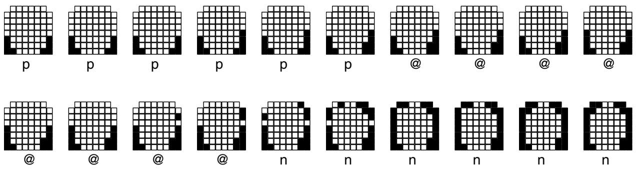

Fig. 1. Typical EPG sequence. Black squares indicate a contact between the tongue and the palate. Segment is from the utterance "The hallway o**pen**s into a huge chamber". Symbol p stands for /p/, @ for /ə/ and n for /n/.

2. THE MOCHA DATABASE

The MOCHA (Multi-Channel Articulatory) [3] 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

The first author is a recipient of a dissertation scholarship through the Operational Programme for Education and Initial Vocational Training ("Heracletus" Project).

1-4244-0469-X/06/$20.00 ©2006 IEEE

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 three 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. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a well-known statistical method with which one creates a data model by projecting his data onto a new set of axes. These axes are the directions in the data space where the data variation is maximum, and are called the *principal components*. The projections of the data are then very close to being uncorrelated among each other. Practically, PCA is accomplished by applying eigenvalue analysis on the data covariance matrix. The eigenvectors are then the principal components.

PCA has been used to explain the variance in EPG data quite succesfully in the past [1].

4. SUPPORT VECTOR REGRESSION

The ϵ-SVR algorithm [4] is a generic supervised learning method that may be used to map a real-valued feature vector to a real output value. Given n training vectors $\mathbf{x_i}$ and a vector $y \in R^n$ such that $y_i \in R$, one wants to find an estimate for the function $y = f(\mathbf{x})$. According to ϵ-SVR, this estimate is:

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

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

$$\text{maximize}$$

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

$$\text{subject to}$$

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

where $C > 0$ and $\epsilon \in (0, 1)$ are parameters chosen by the user.

The kernel function serves to convert the data into a higher-dimensional space in order to account for non-linearities in the estimation function. A commonly used kernel is the Radial Basis Function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \parallel \mathbf{x} - \mathbf{y} \parallel^2), \tag{3}$$

where the γ parameter is selected by the user.

5. DATA PROCESSING

The MOCHA database includes 460 utterances of the fsew0 speaker. In order to render these data into input-output pairs suitable for our purposes, we proceed 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, using HTK, we extract from the speech signal the 12-order Perceptual Linear Predictive Coding Coefficients [5] plus the log energy, using 16ms windows (256 points) with 5ms shifts (this is to match the 200Hz sampling rate of the EPG data). 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 construct input vectors spanning over a large number of acoustic frames. We use context windows of 20 frames (the frame in question plus 10 past ones plus 9 future ones) with a distance of 10 ms between consecutive frames (that is, one out of two of our previously calculated frames). Thus, we have 260-dimensional (20 × 13) acoustic vectors, each spanning a neighborhood of about 200 ms. (In accordance with works on other acoustic-to-articulatory mapping tasks e.g. [6])

From the 460 available utterances the odd-numbered ones will constitute what we will call our "extended training set" and 46 (every 10th utterance beginning with the 6th) will constitute our test set.

6. EXPERIMENTS

As a first step to our method we apply PCA on our extended training set. We keep those principal components with corresponding eigenvalue larger than $1/100$ of the largest eigenvalue. Figure 2 shows a schematic of these 38 principal components.

As a result of PCA, each EPG vector in our data may be decomposed (with a cost of a small loss in reconstruction accuracy, depicted in the top subfigure of Figure 3) as:

$$\mathbf{y_{epg}} = \sum_{i=1}^{38} w_i \mathbf{PC_i}. \tag{4}$$

In this decompositions the $\mathbf{PC}$s remain constant across the dataset, while the ws vary. So, we may introduce a set of mappings from our acoustic vectors (created as described) to

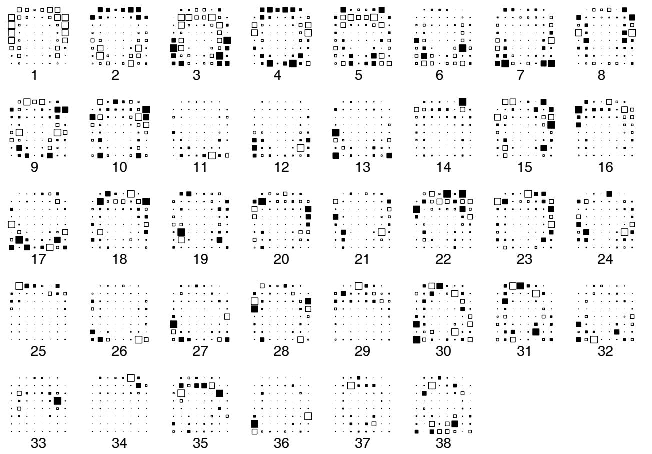

Fig. 2. Principal Components of the EPG data. Each value is represented by a square of size proportional to its absolute value and color black or white whether it is positive or negative.

each of these w-trajectories. Thus, we have 38 distinct, uncorrelated, regression problems, each of which suits the ϵ-SVR algorithm.

Our extended training set includes over 120,000 training examples. This is far too big for SVR to train in a reasonable amount of time. We employ a very simple trick to reduce this amount of data by taking 1 out of 20 examples. Thus, we arrive at a ‘reduced training set’ of about 6,000 examples, which we actually use for SVR training. We choose the RBF kernel with $\gamma = 0.00384615$ and select $C = 1$ and $\epsilon = 0.1$. We use the LibSVM software.

After testing, we measure the Pearson correlation score

$$r = \frac{\sum_i (o_i - \bar{o})(y_i - \bar{y})}{\sqrt{\sum_i (o_i - \bar{o})^2 \sum_i (y_i - \bar{y})^2}}, \qquad (5)$$

between the original (y) and the estimated (o) w-trajectories. The results are shown in the middle section of Figure 3.

The next step is to invert PCA, reverting to the original EPG space. For each EPG electrode and point in time, the output is a real number roughly between -2 and 2. There are two choices: we can either interpret the outputs as *probabilities* that the certain EPG event is a contact or we can make *hard decisions* by assigning negative values as non-contacts and positive ones as contacts. The situation is depicted in Figure 4, where we show the ROC curves [7] (where the activity of each electrode is considered as a separate binary clasification task) for a couple of electrodes in two cases: before and after making the hard decision.

Figure 5 summarizes the results of our method as a function of EPG electrodes. The left subfigure shows the percentage of contacts in the test set, which varies a lot among EPG electrodes. The convention here is that the largest the black square, the closer the percentage is to 100%. Using the same convention, the middle subfigure shows the classification rate achieved at each of the EPG electrodes, *after* making the hard decision. The last subfigure shows the AUCs (that is, the area under the ROC curve, which summarizes the curve into a single number), *before* the hard decision. The bigger the square, the closer the area to 1. There are a few electrodes for which the ROC curve is not defined (the white “cross” in the middle); that is because for those electrodes there is actually no contact at all in the test set (so, in way, the whole classification effort is meaningless). Table 1 presents a very brief summary of the results. We just point out that the overall AUC is *not* just the mean value of the AUCs for individual EPG electrodes. Instead, it is the area under the ROC curve,

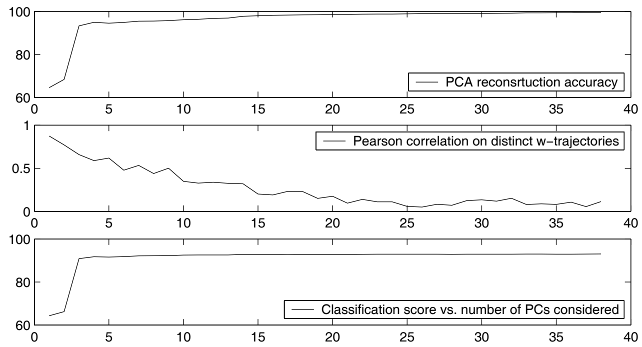

Fig. 3. Top to bottom: EPG reconstruction accuracy as a function of principal components used; Pearson correlation on individual w-trajectories as a function of the corresponding PC; final EPG classification score as a function of the number of PCs considered for the whole classification task.

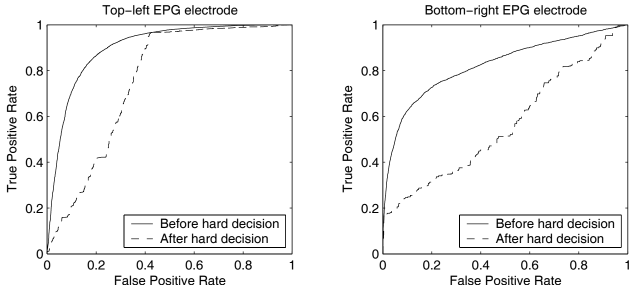

Fig. 4. ROC curves for two EPG electrodes.

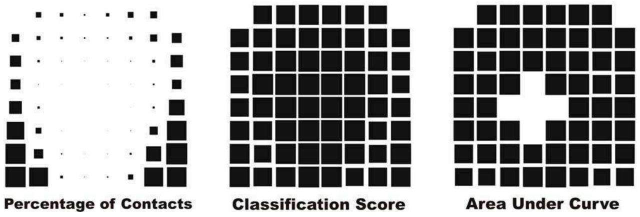

Fig. 5. Results as function of EPG electrodes.

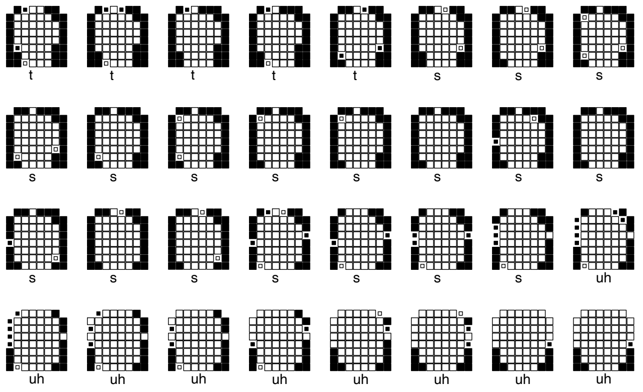

Fig. 6. Detailed classification results. Big black and white squares are contacts and non-contacts respectively correctly classified. Small black squares are contacts classified as non-contacts and small white squares are non-contacts classified as contacts. Segment is form the phrase “Brigh**t** **su**nshine shimmers on the ocean”. Symbol t stands for /t/, s for /s/ and uh for /ʌ/.

plotted from the results of all 62 electrodes treated as a whole. Overall chance level of the test set is 86,28%.

	Min	Max	Overall
Contacts	0%	97,99%	25,99%
Classification Score	81,36%	100%	93,02%
AUC	0,78	0,98	0,93

Table 1. Summary of results. Min and Max represent the minimum and maximum values across the individual EPG electrodes.

Finally, Figure 6 presents the detailed outcome (after hard decision) of testing our final system on a short speech segment.

7. DISCUSSION

We proposed a method for estimating EPG sequences from the speech signal, with results that we believe are encouraging.The proposed system seems to behave much better when its outputs are regarded as probabilities of contacts than hard decisions. At this point, though, we think that binary decisions offer conceptual simplicity of the results.

It may be the case that we used more than enough principal components for the task. The middle subfigure of Figure 3 shows that the performance of SVR seriously degrades while moving to less significant principal components. In the bottom subfigure, we plotted what the system’s performance would be, if we used various numbers of principal components. It seems that after the first 10-12 principal components, overall classification rate reaches a plateau. After all, it is desirable to explain EPG data in terms of as few parameters as possible.

A major drawback of our method is that we used only a small segment of the available data, since training on all data would require quite huge training times. We are now working with mixtures of Support Vectors Regressors, following ideas from [8], which require more reasonable training times, already achieving much better correlation scores on the w-trajectories.

The problem we dealt with includes a heavily structured output space. Dealing with such spaces is a hot issue in recent machine learning literature (e.g. [9]), and there may be methods that better account for the structure of our problem than the method we proposed. We believe that the main strength of our method, is that PCA is already an established tool for the analysis of EPG data.

8. REFERENCES

[1] Miguel Á. Carreira-Perpiñán, *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*, Ph.D. thesis, University of Sheffield, UK, February 2001.

[2] Fiona Gibbon, “Bibliography of electropalatographic studies in English,” Tech. Rep., Queen Margaret University College, Edinburgh, UK, September 2005.

[3] Alan A. Wrench and William J. Hardcastle, “A multichannel articulatory database and its application for automatic speech recognition,” in *5th Seminar on Speech Production: Models and Data*, Kloster Seeon, Bavaria, 2000, pp. 305–308.

[4] Alex Smola and Bernhard Schölkhopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, August 2004.

[5] Hynek Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, April 1990.

[6] Korin Richmond, *Estimating Articulatory Parameters from the Speech Signal*, Ph.D. thesis, The Center for Speech Technology Research, Edinburgh, UK, 2002.

[7] Tom Fawcett, “ROC graphs: Notes and practical considerations for researchers,” Tech. Rep., HP Laboratories, Palo Alto, April 2004.

[8] Ronan Collobert, Samy Bengio, and Yoshua Bengio, “A parallel mixture of SVMs for very large scale problems,” *Neural Computation*, vol. 14, pp. 1105–1114, 2002.

[9] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun, “Support vector machine learning for interdependent and structured output spaces.,” in *ICML*, 2004.