Mapping the Speech Signal onto Electromagnetic Articulography Trajectories Using Support Vector Regression

Asterios Toutios and Konstantinos Margaritis

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

Abstract. We report work on the mapping between the speech signal and articulatory trajectories from the MOCHA database. Contrasting previous works that used Neural Networks for the same task, we employ Support Vector Regression as our main tool, and Principal Component Analysis as an auxiliary one. Our results are comparable, even though, due to training time considerations we use only a small portion of the available data.

1 Introduction

The acoustic-to-articulatory mapping [1, 2], also termed acoustic-to-articulatory inversion of speech, is a special speech processing related problem that has attracted the attention of several researchers for many years now. It refers to the estimation of articulatory (speech production related) information using solely the speech signal as an input source. A succesful solution could find numerous applications, such as helping individuals with speech and hearing disorders by providing visual feedback, very low bit-rate speech coding and the possibility of improved automatic speech recognition.

In the past, the articulatory features used in such a context were mostly inferred by the corresponding acoustic data using vocal-tract models, synthesis models, or linguisting rules. But recent technologies have made it possible to record actual articulator movements in parallel with speech acoustics in a minimally invasive way. This "real" human data is arguably preferable to older techniques, where additional complications may be imposed by intrinsic flaws of the models themselves.

One of the forementioned technologies is the Electromagnetic Misdagittal Articulography (EMMA) or Electromagnetic Articulography (EMA). Roughly speaking, for the aquisition of EMA data, sensor coils are attached to the human subject, on specific places on the lips, the teeth, the jaw, and the soft palate (velum). Then the human subject wears a special helmet that produces an alternating magnetic field that records the position of the coils at end points of small fixed-size time intervals. The outcomes are trajectories that illustrate the movement of the coils. Usually, there are two trajectories for each coil, one for the movement in the front-back direction of the head, and one for the top-bottom direction.

In this paper we follow Richmond's work [1], who proposed a quite succesful mapping of the speech signal to EMA data, using Neural Networks (Multilayer Perceptrons and Mixture Density Networks). We study an alternative –Machine Learning– approach

V. Matoušek et al. (Eds.): TSD 2005, LNAI 3658, pp. 318–325, 2005.
© Springer-Verlag Berlin Heidelberg 2005

using Support Vector Regression, a more recent and very promising method. We also employ, as part of our experimentation, the techinque of Principal Component Analysis, as a means to account for the interrelationships among the EMA trajectories. We use the same dataset as Richmond (though we finally arrive at a significantly smaller training set), namely the fsew0 speaker data from the MOCHA dabase.

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. The articulatory channels include EMA sensors directly attached to the upper and lower lips, lower incisor (jaw), tongue tip (5-10mm from the tip), tongue blade (approximately 2-3cm posterior to the tongue tip sensor), tongue dorsum (approximately 2-3cm posterior from the tongue blade sensor), and soft palate. Two channels for every sensor are recorded at 500Hz: the positioning on the x-axis (front-back direction) and on the y-axis (top-bottom direction).

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 EMA data, as well as the phonemic labels for the fsew0 speaker, a female speaker with a Southern English accent, are used.

3 Support Vector Regression

The ϵ-SVR algorithm [4] is a generalization of the better known Support Vector Classification algorithm [5] to the regression case. Given n training vectors $\mathbf{x_i}$ and a vector $y \in R^n$ such that $y_i \in R$, we want to find an estimate for the fuction $y = f(\mathbf{x})$ which is optimal from a Structural Risk Minimization viewpoint. 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

$$\text{maximize}$$

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

$$\text{subject to } 0 \leq a_i, a_i^* \leq C, i = 1, \ldots, n, \text{ and } \sum_{i=1}^{n} (a_i^* - a_i) = 0. \tag{2}$$

$C > 0$ and $\epsilon \geq 0$ are parameters chosen by the user. The "penalty parameter" C may be as high as infinity, while usual values for ϵ are 0.1 or 0.001.

The "kernel" $k(\mathbf{x_i x_j})$ is a special function which serves to convert the data into a higher-dimensional space in order to account for non-linearities in the estimate 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.

4 Principal Component Analysis

PCA [2] 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 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

$$\mathbf{w_1} = \arg \max_{\|w\|=1} E\{(\mathbf{w}^T \mathbf{x})^2\} \tag{4}$$

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 kth component is:

$$\mathbf{w_k} = \arg \max_{\|w\|=1} E\{\mathbf{w}^T (\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w_i} {\mathbf{w_i}}^T \mathbf{x})^2\}. \tag{5}$$

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 $\mathbf{C}$.

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 function estimation, we process them as follows.

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

Next, we perform a standard Mel Frequency Spectral Analysis [6] on the acoustic signal with the VOICEBOX Toolkit [7], using a window of 16ms (256 points) with a shift of 5ms. 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 *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 17 frames: the frame in question, plus the 8 previous ones, plus the 8 next ones.

The steps taken to process the EMA data are similar to those described by Richmond. First, the EMA data are resampled to match the frameshift of the acoustic coefficients (5ms). At the same time, they are smoothed, using a moving average window of 40ms so that recording noise is eliminated (after all, it is known that EMA trajectories vary relatively slowly with time).

The mean values of the EMA trajectories calculated for every utterance vary considerably during the recording process. There are two kinds of variation: rapid changes, due to the phonemic content of each utterance, and slowly moving trends, mainly due to the fact that te subject's articulation adapts in certain ways during the recording session. It is beneficial to remove from the EMA data the second type of variation, while keeping the first. Thus, we calculate the means, low-pass filter them, and subtract those filtered means from the EMA data. (See Figure 1 for an explanation).

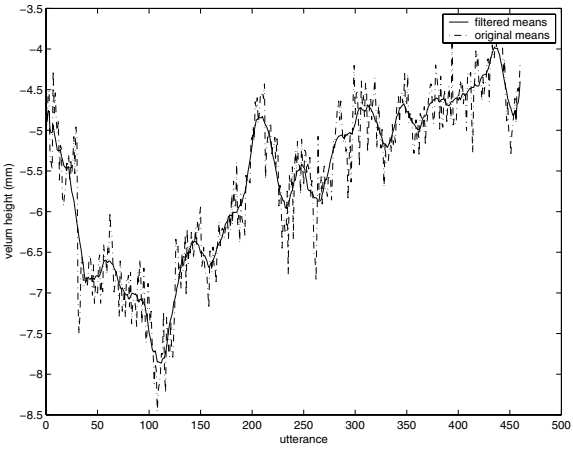

Fig. 1. Mean values of the "velum height" (v_y) channel across the utterances in the recording session. The dashed line shows the real means and the solid line their filtered version which is actually used for normalization.

Finally, we scale the EMA data by four times their standard deviation (across the whole corpus), so that they roughly lie in the interval $(-1, 1)$, something crucial for SVR training.

Thus, we end up with training examples with a 221-dimensional (17×13) real-valued vector as input and a 14-dimensional real-valued vector as output. We split our data into two big halves: the even-numbered utterances constitute a "big training set", and the odd-numbered ones a "big test set". Each one has more than 100.000 examples.

But, since SVR training is a relatively slow process, using the whole "big training set" for training would merely be out of the question. We would like a reduced training set, that is somehow "representative" of the whole corpus. Knowing (from the label files) the phoneme that each of our "big training set" examples corresponds to, we randomly select 200 examples "belonging" to every phoneme. With 44 phonemes in the database, we end up with 8800 training examples.

Finally, for our test set, we simply use 10 utterances spanning across our whole "big test set".

6 SVR Training and Results

The ϵ-SVR algorithm, as described, works for only one output. It does not work "as is" for multiple outputs. Thus, we have to split our problem into 14 distinct (and assumably independent) function estimation problems, considering each time a different EMA trajectory as output.

We use the RBF kernel with $\gamma = 0.0045$ and select $C = 1$, $\epsilon = 0.1$, based on heuristics found in [8], employing the LibSVM software [9] for our experiments. We, finally, virtually "combine" the 14 estimators into one "system".

For evaluating the performance of our system we use two measures. The first one is the RMS error which is an indication of the overall "distance" between two trajectories. It is calculated as:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - t_i)^2} \tag{6}$$

where N is the number of input-output vector pairs, in the test set, o_i is the estimated value for the articulator channel output, and t_i is the real value. The values are rescaled back to the original domain measurement in millimeters.

The second measure is the correlation score, which is an indication of similarity of shape and synchrony of two trajectories. It is calculated by dividing their covariance by the product of their variances:

$$r = \frac{\sum_i (o_i - \bar{o})(t_i - \bar{t})}{\sqrt{\sum_i (o_i - \bar{o})^2 \sum_i (t_i - \bar{t})^2}} \tag{7}$$

where $\bar{o}$ and $\bar{t}$ are the mean channel value for the estimated and real articulator position respectively.

The results of this first experiment are presented in Table 1 and Figure 2.

As a second experiment, and as an attempt to account for the interrelationships between the EMA trajectories we add PCA to the previous experimental context. We know that some pairs of trajectories are highly correlated. By PCA, we move to a new output space where the new trajectories are uncorrelated among each other.

Most of the times PCA is used for data reduction, by "cutting off" components that correspond to small eigenvalues. This is not our case. We just want to render our data into an uncorrelated form, so we keep all 14 Principal Components. We perform SVR, with the exact same parameters as previously, in this new output space and then, at testing, revert back to our original one. Table 2 shows the results of this experiment.

Table 1. Performance of the System of Estimators. (First experiment, without PCA).

Articulator	RMS Error (mm)	Correlation
lower incisor x	1.054	0.479
lower incisor y	1.217	0.807
upper lip x	0.999	0.565
upper lip y	1.327	0.548
lower lip x	1.403	0.499
lower lip y	2.375	0.803
tongue tip x	2.534	0.806
tongue tip y	2.750	0.809
tongue body x	2.339	0.788
tongue body y	2.248	0.814
tongue dorsum x	2.262	0.743
tongue dorsum y	2.573	0.671
velum x	0.455	0.690
velum y	0.397	0.726

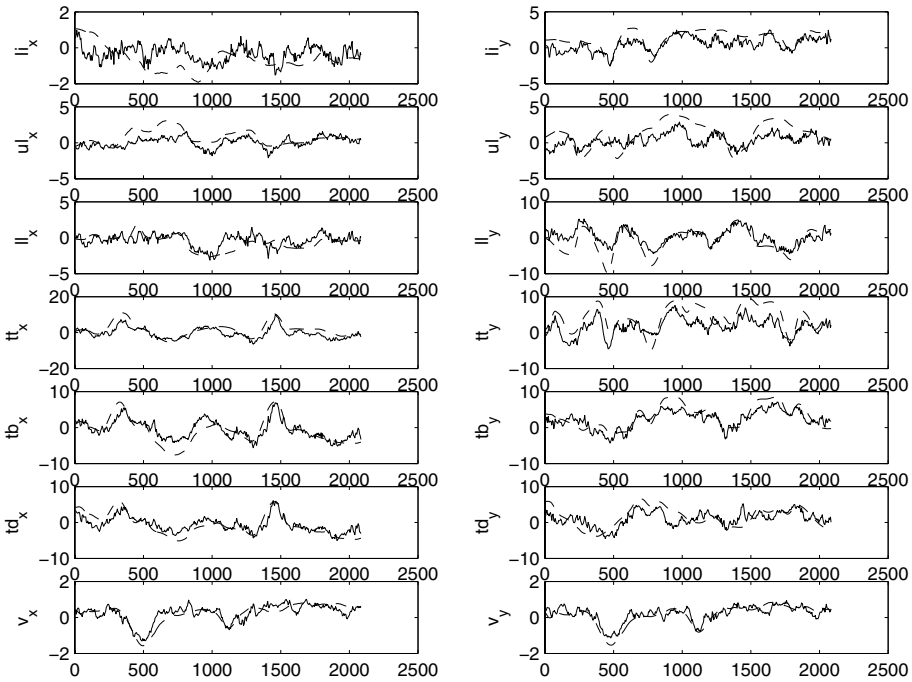

Fig. 2. Real (dashed lines) and estimated (solid lines) articulatory trajectories of fsew0 uttering the phrase "Clear pronunciation is appreciated.". The first column is the projection of the articulator's movement on the x axis and the second on the y axis. From top to bottom: lower incisor (jaw), upper lip, lower lip, tongue tip, tongue dorsum, tongue blade and velum.

Table 2. Performance of the System of Estimators (Second Experiment, with PCA).

Articulator	RMS Error (mm)	Correlation
lower incisor x	1.053	0.481
lower incisor y	1.200	0.812
upper lip x	1.006	0.559
upper lip y	1.327	0.548
lower lip x	1.329	0.550
lower lip y	2.363	0.805
tongue tip x	2.556	0.802
tongue tip y	2.766	0.807
tongue body x	2.353	0.785
tongue body y	2.226	0.818
tongue dorsum x	2.271	0.740
tongue dorsum y	2.557	0.675
velum x	0.452	0.693
velum y	0.399	0.723

7 Conclusion

We applied Support Vector Regression to the task of mapping the acoustic speech signal onto EMA trajectories. Our results were comparable to those found in the literature, even though we used a (selected by a rather ad-hoc procedure) small subset of the data available to us. We extended our method by employing Principal Component Analysis, in order to account for the interrelationships inherent among the trajectories, with a slight increase in performance.

In order to improve further our results we should try to better exploit the vast amount of data in the MOCHA database. This may be done in one of two ways, the first one being to use more training data. Training time is always an issue, but recent findings in the machine learning field, such as Cross-Training [10], seem quite promising in the direction of speeding up things. One second way is to use a more formal way, perhaps by applying a clustering technique to our input space, in order to select training examples.

Finally, PCA lead to only a slight increase in performance. We expected better. It may be the case that other data transformations, such as Independent Component Analysis [11], should also be considered.

References

1. Richmond, K.: Estimating Articulatory Parameters from the Speech Signal. PhD thesis, The Center for Speech Technology Research, Edinburgh, (2002).
2. Carreira-Perpiñán, M.A.: Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction. PhD thesis, University of Sheffield, UK (2001)
3. Wrench, A.A., Hardcastle, W.J.: A multichannel articulatory database and its application for automatic speech recognition. In: 5th Seminar on Speech Production: Models and Data, Kloster Seeon, Bavaria (2000) 305–308
4. Smola, A., Schölkhopf, B.: A tutorial on support vector regression. Statistics and Computing 14 (2004) 199–222

5. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
6. Davis, S.B., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In Waibel, A., Lee, K.F., eds.: Readings in speech recognition. Morgan Kaufmann Publishers Inc. (1990) 65–74
7. Brooks, M.: (The VOICEBOX toolkit) Software vailable at http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.
8. Weston, J., Gretton, A., Elisseeff, A.: SVM practical session (how to get good results without cheating). (Machine Learning Summer School 2003, Tuebingen, Germany.)
9. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
10. Bakir, G., Bottou, L., Weston, J.: Breaking SVM complexity with cross training. In: 18th Annual Conference on Neural Information Processing Systems, NIPS-2004. (2004)
11. Hyvärinen, A., Oja, E.: Independent component analysis: Algorithms and applications. Neural Networks 13 (2000) 411–430