# Learning Articulation from Cepstral Coefficients

*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 work on a special case of the speech inversion problem, namely the mapping from Mel Frequency Cepstral Coeeficients onto articulatory trajectories, derived by EMA. We employ Support Vector Regression, and use PCA and ICA as means to account for the spatial structure of the problem. Our results are comparable to those achieved by older attempts on the same task, indicating probably some natural limitation on the mapping itself.

## 1. Introduction

The acoustic-to-articulatory mapping [1, 2], also termed acoustic-to-articulatory inversion of speech or speech inversion, 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 acquisition 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 which 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 movement in the top-bottom direction.

In this paper we follow Richmond's work [1, 3], who proposed a quite succesful mapping of the speech signal onto EMA data, using Neural Networks. We study an alternative –Machine Learning– approach using Support Vector Regression, a more recent and very promising method. 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) [4] 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 three speakers are available. For the experiments described herein, the acoustic waveform and EMA data, as well as the phonemic labels for the speaker fsew0, a female speaker with a Soutern English accent, are used.

## 3. Mathematical Tools

In this section we brefly describe the mathematical methods we use in this paper, namely $\nu-$Support Vector Regression ($\nu-$SVR), Principal Component Analysis (PCA) and Independent Component Analysis (ICA).

### 3.1. $\nu-$Support Vector Regression

The $\nu$-SVR algorithm [5] is an improvement of the $\epsilon$-SVR algorithm, the generalization of the better known Support Vector Classification algorithm [6] to the regression case. Given $l$ training vectors $\mathbf{x_i}$ and a vector $y \in R^l$ such that $y_i \in R$, one wants to find an estimate for the function $y = f(\mathbf{x})$ which is optimal from a Structural Risk Minimization viewpoint. According to $\nu$-SVR, this estimate is:

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

where $b$ is a bias term and $k(\mathbf{x_i x_j})$ is a special function called the *kernel*. The coefficients $a_i$ and $a_i^*$ are the solution of the

quadratic problem

$$\text{maximize}$$

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

$$\text{subject to}$$

$$\sum_{i=1}^{l}(a_i - a_i^*) = 0$$

$$\sum_{i=1}^{l}(a_i - a_i^*) \leq C\nu l$$

$$a_i, a_i^* \in [0, C] \tag{2}$$

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 $\gamma$ parameter is selected by the user.

### 3.2. Principal Component Analysis

With PCA [2] 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.

### 3.3. Independent Component Analysis

ICA [7] is a relatively recently developed method in which the goal is to find a linear representation of nongaussian data so that the components are statistically independent, or as independent as possible. What underlies ICA is an idea of inverting the central limit theorem: When various independent variables are mixed, the net distribution more or less approximates normal distribution. So, when searching for the original, unmixed signals, one can search for maximally non-normal projections of the data distribution.

There are various algorithms that accomplish ICA. One of the most prevalent is the FastICA algorithm, which is used herein. Two preprocessing steps are necessary before applying ICA: the first one is PCA and the second is whitening. The latter is a simple transformation that renders the covariance matrix of the data into an identity matrix.

The output of ICA is two matrices: the *mixing matrix* $\mathbf{A}$ and the *separating matrix* $\mathbf{W}$ so that

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad \text{and} \quad \mathbf{s} = \mathbf{W}\mathbf{x} \tag{4}$$

where $\mathbf{x}$ are the original data and $\mathbf{s}$ the transformed data, or *independent components*.

## 4. Data Processing and Training Set Selection

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.

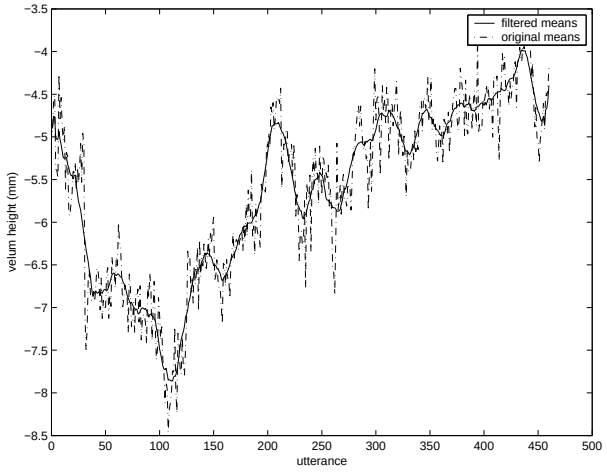


Figure 1: Mean values of the "velum height" ($v_y$) channel across the utterances in the recording session. The dashed line shows the real means and the solid line their filtered version which is actually used for normalization.

First, based on the label files we omit silent parts from the beginning and end of the utterances. During silent stretches the articulators 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, 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 the 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 the 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).

So, we end up with training examples, each consisting of a 221-dimensional ($17 \times 13$) 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 what

we call an "extended training set", and the odd-numbered ones an "extended test set". Each one contains more than 100.000 examples.

But, since SVR training is a relatively slow process, it would be far too much time consuming to use our whole "extended training set" for training. In order have a more compact training set, yet reperesenative of the whole corpus, we employ a simple trick: we randomly select 200 training examples corresponding to every one of the 44 different phonemic labels in the MOCHA database, thus arriving at a training set consisting of 8800 examples.

## 5. SVR Training and Results

The $\nu$-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 function estimation problems, considering each time a different trajectory, or channel, as output. We actually follow three approaches: In the first one we consider the 14 original EMA trajectories (processed as described so far) as the outputs of our set of function estimators (Plain SVR approach). In the second one we transform those trajectories by applying PCA (PCA + SVR approach), and in the third one by applying ICA (ICA + SVR approach).

In both the PCA and ICA cases we could possibly reduce the dimensionality of our data removing the less important principal or independent components, respectively. We don't do that, we keep all 14 components. Our reason for applying this transformation is to account for (and hopefully remove) the *spatial interrelationships* inherent in the EMA data.

Just before SVR training we perform two further preprocessing steps on our output data. Firstly we *center* the data so that the mean value of every channel is zero, and, secondly we scale the data by four times their standard deviation, so that they roughly lie in the interval $(-1, 1)$, something crucial for SVR training.

In order to train our function estimators, we use the RBF kernel with $\gamma = 0.0045$ and select $C = 1$ and $\nu = 0.4$, based on a combination of several heuristcs and small scale experiments of ours. We employ the LibSVM software [9] for our experiments.

After testing our estimators we invert the processes of scaling and centering. For our two latter approaches we invert PCA or ICA, respectively.

Finally, we smooth the output trajectories using again a moving average window of 40ms. In a way, this compensates for the fact that by no other means do we account for the *temporal interrelationships* in the EMA data (mainly, the physiological fact that the trajectories vary relatively slowly in time).

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:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - y_i)^2} \tag{5}$$

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 $y_i$ is the real value.

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

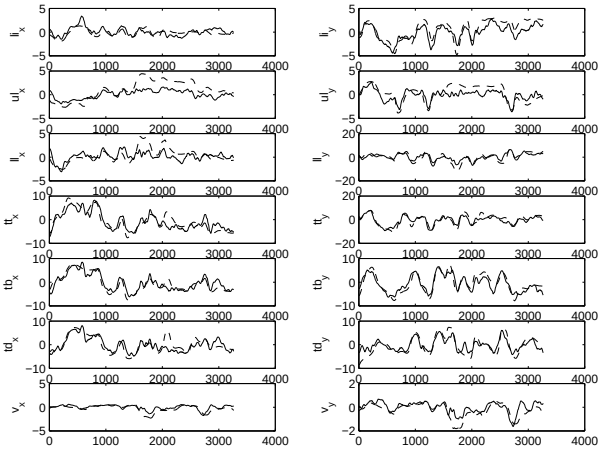

Figure 2: Real (dashed lines) and estimated by plain SVR (solid lines) articulatory trajectories of fsew0 uttering the phrase "The jaw operates by using antagonistic muscles", plotted against time in milliseconds. From top to bottom: lower incisor (jaw), upper lip, lower lip, tongue tip, tongue dorsum, tongue blade and velum. The first column shows the projections of the articulators' movement on the $x$ axis, while the second one those on the $y$ axis.

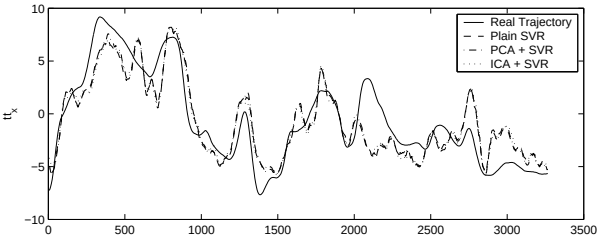


Figure 3: Projection of the movement of the tongue tip on the $x$ axis, while fsew0 is uttering the phrase "The jaw operates by using antagonistic muscles", and the corresponding estimated trajectories by the three methods presented in this paper

of their variances:

$$r = \frac{\sum_i (o_i - \bar{o})(y_i - \bar{y})}{\sqrt{\sum_i (o_i - \bar{o})^2 \sum_i (y_i - \bar{y})^2}} \tag{6}$$

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

For testing our system we use 10 utterances spanning our whole "extended test set". Our overall results for this test set are presented in Table 1 where we also include the standard deviation of the EMA cannels (as an indication of the range of their values), and the results achieved in [3], for a comparison.

Figure 2 shows the real and estimated trajectories for a single utterance using the Plain SVR approach. Figure 3 shows the real and estimated trajectories for a single EMA channel using all three approaches.

| | | **Plain SVR** | | **PCA + SVR** | | **ICA + SVR** | | **MLP** *(adapted from [3])* | |
|---|---|---|---|---|---|---|---|---|---|
| **EMA Channel** | $\sigma$ | $E_{RMS}$ | $r$ | $E_{RMS}$ | $r$ | $E_{RMS}$ | $r$ | $E_{RMS}$ | $r$ |
| Lower incisor x | 1,156 | 1,044 | 0,473 | 1,044 | 0,469 | 1,049 | 0,461 | 0,89 | 0,56 |
| Lower incisor y | 2,068 | 1,130 | 0,840 | 1,121 | 0,844 | 1,123 | 0,845 | 1,19 | 0,80 |
| Upper lip x | 1,155 | 0,946 | 0,616 | 0,948 | 0,614 | 0,942 | 0,623 | 0,99 | 0,58 |
| Upper lip y | 1,626 | 1,277 | 0,592 | 1,277 | 0,595 | 1,273 | 0,596 | 1,16 | 0,72 |
| Lower lip x | 1,655 | 1,348 | 0,546 | 1,355 | 0,539 | 1,345 | 0,548 | 1,21 | 0,61 |
| Lower lip y | 4,088 | 2,228 | 0,838 | 2,236 | 0,836 | 2,248 | 0,835 | 2,73 | 0,75 |
| Tongue tip x | 3,974 | 2,387 | 0,832 | 2,383 | 0,832 | 2,385 | 0,831 | 2,43 | 0,79 |
| Tongue tip y | 4,715 | 2,531 | 0,839 | 2,527 | 0,841 | 2,529 | 0,841 | 2,56 | 0,84 |
| Tongue body x | 3,709 | 2,206 | 0,815 | 2,210 | 0,814 | 2,205 | 0,815 | 2,19 | 0,81 |
| Tongue body y | 4,104 | 2,101 | 0,839 | 2,130 | 0,834 | 2,132 | 0,834 | 2,14 | 0,83 |
| Tongue dorsum x | 3,306 | 2,153 | 0,772 | 2,159 | 0,771 | 2,164 | 0,769 | 2,04 | 0,79 |
| Tongue dorsum y | 3,388 | 2,478 | 0,697 | 2,461 | 0,702 | 2,482 | 0,695 | 2,31 | 0,71 |
| Velum x | 0,641 | 0,433 | 0,719 | 0,433 | 0,720 | 0,434 | 0,718 | 0,42 | 0,79 |
| Velum y | 0,612 | 0,370 | 0,761 | 0,367 | 0,764 | 0,369 | 0,760 | 0,41 | 0,77 |

Table 1: Overall performances on the test set of the system of function estimators derived by the three approaches. Also presented: the standard deviations of the dataset and the results achieved in [3]

## 6. Conclusion

In [1, 3], Richmond trains an MLP in order to estimate EMA trajectories. He uses a slightly different parametrization of the acoustic signal from the one we present, employing filterbank analysis (which is nevertheless very closely related to Mel Frequency Cepstral Analysis) and arriving at 400-dimensional input vectors. One of the main motivations for the work presented here was to improve upon Richmond's results, by using a more recent regression method. From Table 1, it is clear that this goal wasn't achieved. Instead, we did a little better at some EMA channels, a little worse on some others, with the whole pictures being quite similar.

Of course, we might excuse ourselves by reminding that, due to training time considerations, we used only a small subset of the data available to us. Indeed, we used 8800 training examples, while Richmond used more than a hunderd thousands. We selected these training examples by a rather ad hoc procedure, depending on the phomenic labels. The use of a more structured approach, e.g. clustering, to this selection, might be a step forward.

We applied PCA and then ICA, in an attempt to account for the spatial structure of the problem, since, apparently, the EMA trajectories must be heavily dependent upon each other. To our dissapointment, this didn't improve the situation (the curves in Figure 3 are almost identical) , indicating that these spatial interrelationships are not really the problem.

Despite everything, what makes us sceptical is the similarity between our results and Richmond's ones. Perhaps, it is the case that there is an upper, natural limit, to the quality of the mapping between MFCCs or filterbank coefficients and the EMA trajectories, that cannot be exceeded, regardless the mathematical method used.

So, an obvious future work direction, towards the goal of better estimating EMA trajectories, would be to consider different kinds of speech signal paramterization, such as LPC or PLP. Another thing is to try to account for the temporal structure of the problem, in the sense that the value of an EMA channel at one time istant relies heavily on the values at previous instants, and it would be beneficial if information on past EMA values could somehow be supplemented to acoustic information. Nevertheless, this is a difficult problem.

## 7. Aknowledgement

The first author is supported by the EPEAEK 2–Heracletus Dissertation Scholarship, code 237-88724-2.

## 8. References

[1] Korin Richmond. *Estimating Articulatory Parameters from the Speech Signal*. PhD thesis, The Center for Speech Technology Research, Edinburgh, 2002.

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

[3] Korin Richmond, Simon King, and Paul taylor. Modelling the uncertainty in recovering articulation from acoustics. *Computer Speech and Language*, 17:153–172, 2003.

[4] 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*, pages 305–308, Kloster Seeon, Bavaria, 2000.

[5] Alex Smola and Bernhard Schölkhopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[6] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[7] Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13, 2000.

[8] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In Alexander Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*, pages 65–74. Morgan Kaufmann Publishers Inc., 1990.

[9] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/˜ cjlin/libsvm.