Estimating Tongue-Palate Contact Patterns from the Speech Signal

Asterios Toutios and Konstantinos Margaritis

Parallel and Distributed Processing Laboratory, Department of Applied Informatics
University of Macedonia, 156 Egnatia Str., 54006, Thessaloniki, Greece
{*toutios, kmarg*}*@uom.gr*

Abstract

Electropalatography (EPG) is a technique that determines the contact patterns between the tongue and the hard palate during speech. In one of its most common forms, it utilizes an artificial palate with 62 silver electrodes embedded in its tongue-facing surface. At small regular time intervals it is recorded whether a specific electrode is contacted or not by the tongue, leading to tongue-palate contact patterns.

EPG is nowadays a relatively well-estabilshed tool in phonetic research, in the clinical treatment of people with articulation difficulties or cleft palate, and also in the teaching of second languages. Still, the derivation of EPG data is a rather expensive and difficult process. What is suggested herein is that a means of estimating EPG patterns directly from the acoustic speech signal - with no need of any special equipment - would be of great value to speech pathologists and phoneticians alike.

This paper presents work towards finding a mapping between acoustic parameters, namely the Mel Frequency Cepstral Coefficients, derived directly from the speech signal, and the corresponding EPG patterns. It may be regarded as a special case of a more general problem called acoustic-to-articulatory inversion, or speech inversion, which refers to finding mappings between the speech signal and some kind of articulatory parameters. One of the main motives for this research field is that the additional articulatory information could be used to improve the performance of current speech recognition systems. EPG patterns could also be used in such a context.

For a solution of the problem described, we investigate the utilization of Support Vector Machines, a relatively new and very promising supervised learning technique, of which not many applications have yet appeared in the speech processing field.

The source of the data we use is the MOCHA database, which is well documented and publicly available via the Web, thus allowing for comparisons of other researcher's results to ours.

1. Introduction

Electropalatography (EPG) is a widely used technique for recording and analysing one aspect of tongue activity, namely its contact with the hard palate during continuous 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 62 electrodes exposed to the lingual surface. When contact occurs between the tongue surface and any of the electrodes, a signal is conducted to an external processing unit and recorded. This process leads, at discrete time instants, to tongue-palate contact patterns like the one shown in Figure 1

	x	x	x	x	x	x	
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	o	o	x	x	x
x	x	o	o	o	o	x	x
x	x	o	o	o	x	x	x
x	x	o	o	o	x	x	x
x	x	x	o	o	x	x	x

Figure 1: EPG instance. x indicates a contact point between the tongue and the hard palate, while o indicates a non contact point.

EPG is nowadays an important tool in speech and language therapy, especially in the treatment of articulation disorders associated with cleft palate [5]. It is used in phonetic research in a variety of contexts (e.g. [7]). It has also been suggested that visual feedback from EPG might be helpful for second language acquisition.

Nevertheless, only few speech therapists or phoneticians have actual access to the relatively specialized equipment needed for the derivation of EPG data. One of the things suggested here is that some means of estimating EPG patterns straight from the acoustic speech signal would be of much interest.

We are exploring the, hopefully existing, mapping between acoustic parameters computed from the speech signal by means of signal processing techniques and the corresponding EPG patterns. We are utilizing stochastic modeling and machine learning techniques. To this end we make use of a certain amount of speech/EPG examples.

For the purposes of this paper we use the Mel Frequency

Cepstral analysis for the parametrization of the speech signal. By making the assumption that every EPG event (a contact or a non-contact at a certain electrode and point in time) depends only on the speech signal and is *independent* of other EPG events (i.e. concurrent activations of neiboughring electrodes, or previous activations of the same electrode), the problem of estimating EPG patterns ends up being a fairly straightforward two-class pattern recognition problem. For its solution, we adopt herein a Support Vector Machine (SVM) framework.

Our work may be considered in the context of *acoustic-to-articulatory inversion*, a field that has risen the attention of several reseracher s [12]. This refers to the general problem of mapping the acoustic speech signal onto a space describing the configuraton of the human vocal tract that actually produced this signal. The information derived this way may be used towards an improvement of the performance of current automatic speech recogntion systems. Several techniques that combine acoustic and articulatory features in a speech recognition context have been indeed proposed [9, 11].

The next section is a brief introduction to the MOCHA database, the source of our data, with emphasis on its EPG part. Section 3 is rough presentation of the basic SVM framework for pattern recognition, and one of its variants for dealing with unbalanced datasets (which is the case for EPG data). In Section 4 we explain the way we parametrize the speech signal and more generally process our data. In Section 5 we explain our strategy for selecting parameters for our SVM models. We present our results in Section 6, and finally draw conclusions in the last section.

2. The MOCHA Database

The MOCHA (Multi-Channel Articulatory) database is evolving in a purpose built studio at the Edinburgh Speech Production Facility at Queen Margaret University College [15].

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. Support Vector Machines

A Support Vector Machine (SVM) is a supervised learning technique for pattern recognition. An SVM is a maximum-margin hyperplane that lies in some space. Given training examples labeled either positive or negative, a maximum-margin hyperplane splits the positive and negative training examples, such that the distance from the closest examples (the margin) to the hyperplane is maximized.

A thorough presentation of the SVM framework for pattern recognition is beyond the scope of this paper. The interested reader is referred to [2] as a starting point. What follows is the formulation of the algorithm used in our experiments, namely the C-Support Vector Classification (C-SVC) [13].

Given training vectors $x_i \in R^n,\ i = 1, \ldots, l$ in two classes, and a target vector $y \in R^l$ such that $y_i \in \{1, -1\}$, C-SVC, solves the primal problem:

$$\begin{gathered}\text{minimize}\\ \frac{1}{2}w^T w + C\textstyle\sum_{i=1}^{t}\xi_i\\ \text{subject to}\\ y_i(w^T\Phi(x_i) + b) \geq 1 - \xi_i\\ \xi_i \geq 0,\ i = 1, \ldots, l.\end{gathered} \tag{1}$$

Its dual is

$$\begin{gathered}\text{minimize}\\ \frac{1}{2}a^T Q a - e^T a\\ \text{subject to}\\ 0 \leq a_i \leq C,\ i = 1, \ldots, l\\ y^T a = 0,\end{gathered} \tag{2}$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an l by l positive semidefinite matrix $Q^{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) \equiv \Phi(x_i)^T\Phi(x_j)$ is the kernel. Here, training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function Φ, giving thus rise to a machine that is, in general, non-linear on the data.

The decision function of such a machine is

$$sign(\sum_{i=1}^{l} y_i a_i K(x_i, x) + b). \tag{3}$$

The choice of the kernel function $K(x_i, x_j)$ is quite important for the implementation of an SVM. Though new kernels are often being proposed by researchers the most common ones are the linear kernel

$$K(x_i, x_j) = x_i^T x_j; \tag{4}$$

the polynomial kernel

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d,\ \gamma > 0; \tag{5}$$

the radial basis function (RBF) kernel

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2),\ \gamma > 0; \tag{6}$$

and the sigmoid kernel

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r), \tag{7}$$

where γ, r and d are kernel parameters.

For a relatively small amount of training data, Problem 2 can be solved with any general purpose optimization package that solves linearly constrained convex quadratic problems. However, for larger problems, there is a need for a decomposition algorithm so that only portions of the training data will be handled at a given time. For the experiments described herein, we use the LIBSVM library for Support Vector Machines [3] which implements such an algorithm.

3.1 Weighted SVM

In the case where there is an unequal proportion of data between the two classes we may need a *weighted* variant of the C-SVC algorithm [8]. That is

$$\begin{gathered}\text{minimize}\\ \frac{1}{2}\|w\| + C^+ \textstyle\sum_{i:y_i=+1} \xi_i + C^- \sum_{i:y_i=-1} \xi_i\\ \text{subject to}\\ y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i,\\ \xi_i \geq 0, i = 1, \ldots, l.\end{gathered} \tag{8}$$

A usual heuristic is that

$$C^+/C^- = l^-/l^+ = \alpha, \tag{9}$$

where $l^+ = \sum_{i:y_i=+1} 1$ and $l^- = \sum_{i:y_i=-1} 1$.

We would like to find a value for the penalty parameter, call it C^* so that the expressions 1 and 8 are equivalent. We have

$$C^* \sum_{i=1}^{l} \xi_i = C^+ \sum_{i:y_i=+1} \xi_i + C^- \sum_{i:y_i=-1} \xi_i \tag{10}$$

If we assume that ξ_i=const. (which not actually true but reasonable for our purpose), we get

$$\begin{aligned}(l^+ + l^-)C^* &= l^+C^+ + l^-C^-\\ (1+\alpha)C^* &= C^+ + \alpha C^-\end{aligned} \tag{11}$$

Combining Equations 9 and 11 we get

$$C^- = \frac{1+\alpha}{2\alpha} C^*,$$

and

$$C^+ = \frac{1+\alpha}{2} C^*.$$

4. Data Processing

Several processing steps are carried out in order to render the acoustic and EPG data into a format suitable for use with SVMs.

First, based on the label files, silent parts from the beginning and end of the 460 fsew0 uttereances are omitted. This is necessary since, during silent stretches, the tongue can potentially take any configuration, something that could pose serious difficulties to SVM training.

Next, Mel Frequency Cepstral Analysis [4] is carried out on the acoustic signal, using a Hamming window of 10ms with a shift of 5ms. (These values are chosen so as to have a perfect one-to-one match between the acoustic frames and the EPG data, sampled at 200Hz.) Including the 0'th order coefficient, 19 Mel Frequency Cepstral Coefficients (MFCCs) plus the log energy are derived. The VOICEBOX Toolkit [1] is used to this end.

The classifier we are looking for would have to account for the dynamic properties of the speech signal. Since this situation cannot be (at least to the knowledge of the writers) *explicitly* dealt with in the SVM context, we adopt a commonplace *spatial metaphor* for time. This literally means that instead of using parameters from the frame in question alone, we need to construct larger input vectors that contain additional parameters from previous frames. Nevertheless, considerations on the SVM training time don't allow us to use too many such previous frames. So, for every time-frame of speech a vector of acoustic parameters is constructed containing the coefficients and log energy of the frame in question plus the four previous ones.

What we end up with is, for every time-frame of real speech, a $\mathbf{x}, \mathbf{y}$ pair, where $\mathbf{x}$ is a 100-dimensional (5×20) vector of acoustic parameters, and $\mathbf{y}$ is a 62-dimensional EPG vector comprising of $\{+1, -1\}$ values, one for each elctrode where a contact or non-contact between the tongue and the hard palate may be recorded.

From the 460 utterances of the fsew0 speaker contained in the database, 368 are included in the training set and 46 form the test set. Another 46 utterances form a separate validation set, which will be of no actual use for the experiments described herein.

5. SVM Model Selection

As already mentioned, we treat every point of contact between the tongue and the palate independently of the others. This approach leads to 62 different problems with the exact same properties as the problem described in Section 3, and actually means that we are going to train 62 different support vector machines, one for every point. Nevertheless, we should expect that the problems have some common properties, and so we will adopt a common model selection strategy for all of them. Figure 2 which shows a numbering of the contact points should be a useful refernce for the following.

	1	2	3	4	5	6	
7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38
39	40	41	42	43	44	45	46
47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62

Figure 2: Numbering of (contact) points

A model selection strategy refers to the choice of a kernel, the corresponding kernel parameters and the "penalty parameter" C of Equation 2. Such is needed, since the Support Vector Machines framework as described in Section 3 gives plenty of choices.

A first question is the choice of kernel. It is suggested in the literature [6] that in general the RBF kernel is a reasonable first choice. The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore the linear kernel is a special case of RBF and also the sigmoid kernel behaves like RBF for certain parameters.

The second reason for choosing the RBF kernel is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel.

Finally the RBF kernel has less numerical difficulties. One key point is that $0 < K(x_i, x_j) \leq 1$ in contrast to polynomial kernels of which kernel values may go to infinity ($\gamma x_i^T x_j + r > 1$) or zero ($\gamma x_i^T x_j + r < 1$) while the degree is large. Moreover, we must note that the sigmoid kernel is not valid (i.e. not the inner product of two vectors) under some parameters.

The second question is which penalty parameter C and kernel parameter γ we should use for our task. The adopted procedure is called grid-search using cross validation and is as follows:

Our training set as described consists of 196266 $\mathbf{x}, \mathbf{y}$ pairs. We take a subset of them by taking one out of 50 pairs, ending up with 3924 training examples. We then divide this "cross-validation set" into four equal sized subsets. For pairs of parameters C and γ, we train and test four differnet classifiers using each of these subsets as a test set and the remaining three as a training set, calculating finally the mean classification accuracy of the four of them (called the "cross-validation accuracy").

The goal is to identify the pair of parameters (C, γ) that gives the best cross-validation accuracy. We first try a coarse grid search, trying exponentially growing sequences of C and γ ($C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$). After identifying a "better region" on the grid, we conduct a finer grid search on that region.

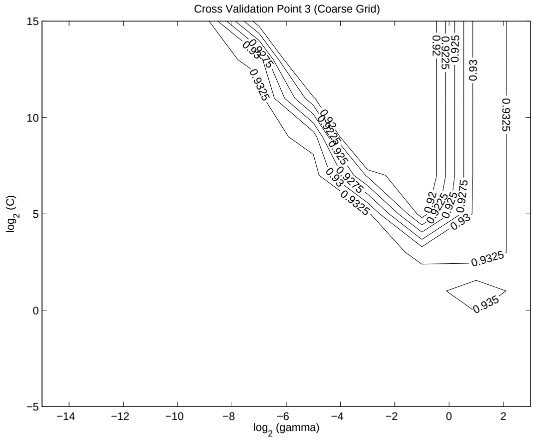

Figure 3: Coarse grid search for point 03

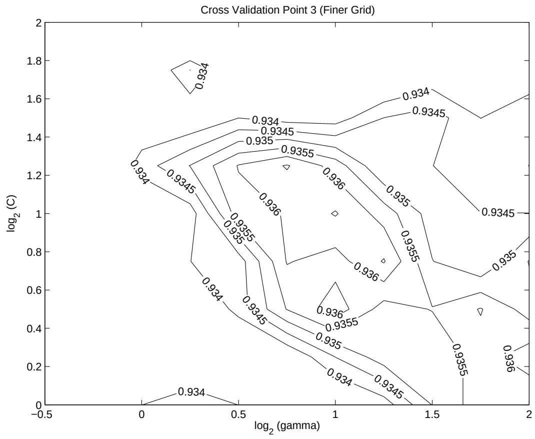

Figure 4: Finer grid search for point 03

As already mentioned, we are about to train 62 different support vector machines, one for every possible contact point between the tongue and the hard palate. Since it would be too much time-consuming to perform cross-validation for every point, we adopt a slightly different strategy. Assuming that, since the properties of the problems are similar, the final (C, γ) pairs will also be similar, we perform full cross-validation *on some points*, in order to subsequently select a (C, γ) pair that would provide a fairly good cross-validation accuracy for all of them. Figures 3, 4, 5, 6, 7, 8, 9 and 10 show cross-validation results for points 3, 15, 37 and 48.

For point 3 the best cross-validation accuracy is obtained at $(C = 2^1, \gamma = 2^1), (C = 2^{0.75}, \gamma = 2^{1.25})$ and $(C = 2^{1.25}, \gamma = 2^{0.75})$; for point 15 at $(C = 2^{-0.5}, \gamma = 2^{0.25})$; for point 37 at $(C = 2^{0.75}, \gamma = 2^{-0.5})$ and for point 48 at $(C = 2^0, \gamma = 2^1), (C = 2^{0.75}, \gamma = 2^0)$ and $(C = 2^1, \gamma = 2^{-0.25})$. The results are not identical, so we have to chose a (C, γ) pair that gives good cross-validation *on average*, to

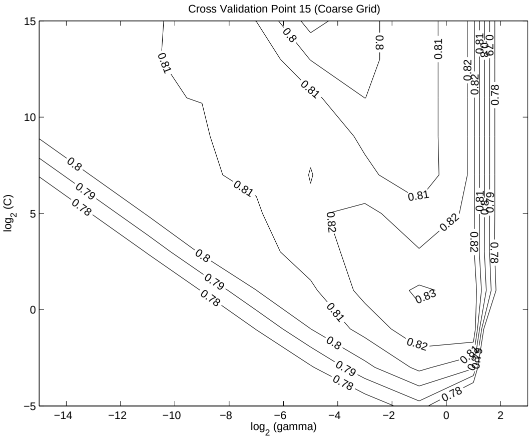

Figure 5: Coarse grid search for point 15

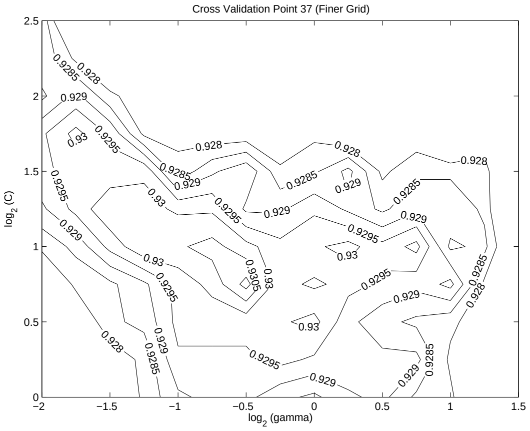

Figure 8: Finer grid search for point 37

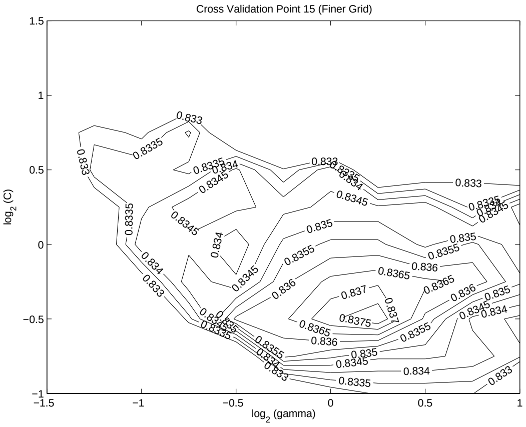

Figure 6: Finer grid search for point 15

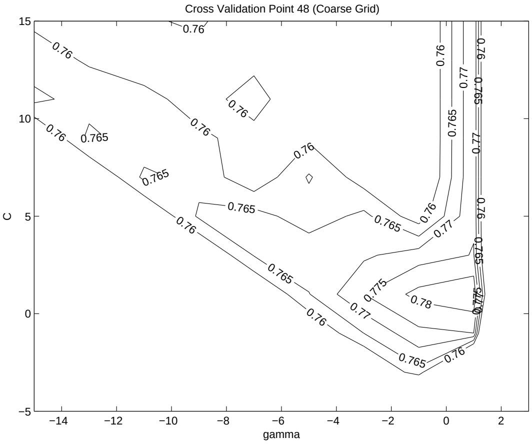

Figure 9: Coarse grid search for point 48

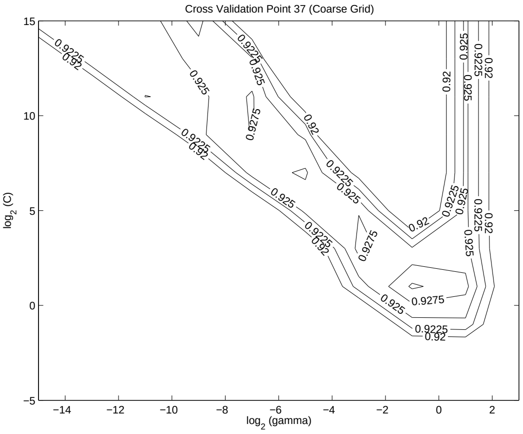

Figure 7: Coarse grid search for point 37

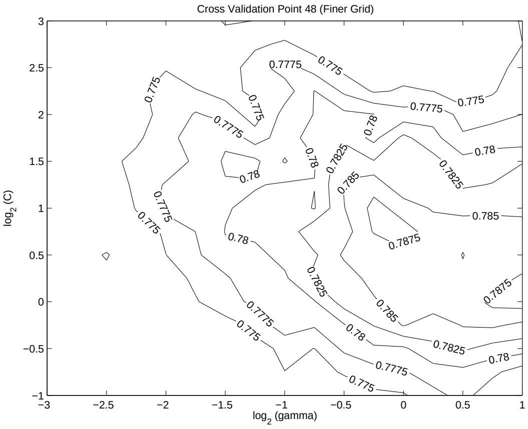

Figure 10: Finer grid search for point 48

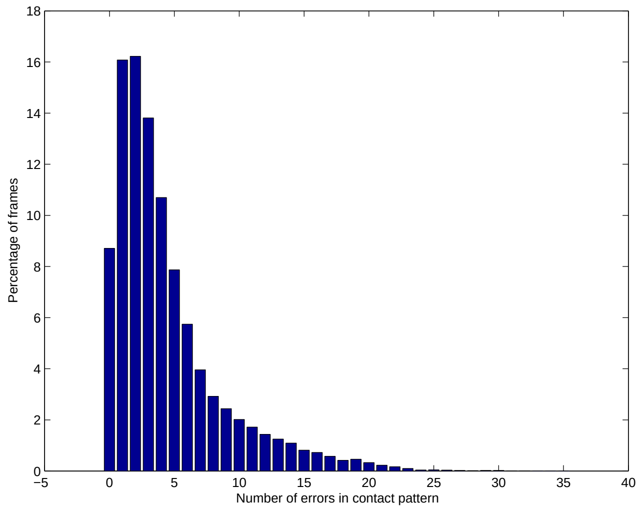

Figure 11: Distribution of errors across frames

use for our task in hand. This would be $(C = 2^{0.5}, \gamma = 2^0)$.

After the model selection we are ready to proceed to our actual task in question, namely the training and testing of the 62 classifiers, each for every possible contact point. The training set of 196266 training examples we have is too big for the SVMs to be trained in a reasonable amount of time, so we do some data reduction by taking one out of five examples ending up with 39253 input-output pairs. Our test set consists of 25514 examples.

6. Results

Our results are presented in the first part of Table 1(under *plain SVM*) , where, for each point we show, the percentage of contacts and non-contacts (positive or negative examples) in our final test set (the corresponding numbers for the training set are similar) the classification rate of the final classifiers; the percentage of contacts which are classified correctly by the classifier; and the percentage of non-contacts which are classified correctly.

For a few points (e.g. 26, 34) we observe a 100% classification rate, which is quite plasmatic, since in the test set only negative examples are observed (i.e. no contacts exist in the test set and there is no actual use of a classifier). Other points (e.g 18, 27, 41, 50) show a very high classification rate which, nevertheless, accounts only for the fact that the chance level of the data (defined as the maximum percentage of the two categories occuring in the test set) is very high. The two last columns of Table 1 indicate that the classifier does nothing but assigning the most probable output to all the test examples. On the contrary, at points like 11, 24 and 54, we observe a non-zero (even though still small) recognition rate on the sparse category , which is an encouraging fact, since it indicates a capability on part of the classifier to perform some actual classification.

For the majority of possible contact points the sets remain realtively non-balanced, and recognition rates for the two categories display remarkable difference between them. Nevertheless, with only one exception (point 51), the final classification rates are higher than the corresponding chance levels. On the other hand, for the points for which the training and test sets are fairly balanced (e.g 13, 22, 38, 45) the classifiers perform quite well, with classification rates $> 80,99\%$, in general far bigger than the corresponding chance levels.

Figure 11 shows another aspect of the results, namely the distributions of the errors made by the set of the SVMs per contact pattern (or time frame). For most of the possible time frames, only few (≤ 3) errors are made, nevertheless there exist a few frames with a large number of errors (up to 32). The average of this distribution is $\sim 4,4$ errors per frame.

It is apparent that one of the drawbacks of the approach just described derives from the fact that, in our data, it is common that the one of the two classes (usually the “non-contacts”) has much more examples than the other. This causes problems, in the sense that our classifiers are biased, performing far better on the “compact” category than on the sparse one. In order to alleviate for this situation we attempt to make use of the weighted SVM, as described in section 3.1, with $C^* = 2^{0.5}$. The results are shown in the second part of Table 1.

Inspection of the last two columns of the table reveals that the weighted SVM achieves the expected effect: the recognition rates on the two categories are much closer to each other than in the plain SVM case, sometimes almost equal. Nevertheless, the recognition rate of the “compact” category decreases, leading inevitably to a lower overall classification rate.

7. Conclusion

We have presented experiments on the acoustic-to-electropalatogaphic inverse mapping with Support Vector Machines, using the fsew0 speaker data from the MOCHA database. We have achieved an overall classification rate of $92,90\%$. This is no reason to overjoy, since the data expose a relatively high chance level. Further experiments with a weighted variant of the typical SVM did not give rise to a clear improvement.

One of our original questions was wether the Mel Frequecy Cepstral Analysis is a parametrization tool well-suited for our given problem, as it is for other speech recognition related tasks. This is still an open question. Other kinds of parametrization could also be considered.

We have used the Radial Basis Function kernel for our SVMs. It is clearly suggested in the literature that this is the best kernel *to start with*. Other kernels could possibly give rise to better results. In order to select training parameters we performed cross-validation experiments on a few EPG points, took a pair of parameters that seemed to do well on all of these points, and applied them to *all* of our points for

	Data Statistics		*Plain SVM*			*Weighted SVM*		
Point	Positive Examples	Negative Examples	Class/tion Rate	Rate of Positives	Rate of Negatives	Class/tion Rate	Rate of Positives	Rate of Negatives
1	25,79	74,21	88,28	68,04	95,31	85,70	86,67	85,36
2	17,92	82,08	88,22	52,39	96,05	83,03	85,74	82,44
3	7,36	92,64	93,74	21,25	99,50	87,08	75,56	87,99
4	15,63	84,37	88,24	44,67	96,31	81,87	84,88	81,31
5	28,61	71,39	88,93	73,68	95,04	86,54	88,95	85,57
6	35,52	64,48	85,88	72,25	93,39	85,06	83,87	85,71
7	38,31	61,69	85,12	72,78	92,78	84,86	82,91	86,06
8	17,61	82,39	87,89	46,19	96,80	80,94	84,96	80,08
9	10,08	89,92	91,98	32,09	98,69	85,95	83,03	86,39
10	4,90	95,10	95,11	1,28	99,95	89,12	66,35	90,29
11	4,56	95,44	95,46	0,52	100,00	89,67	62,29	90,98
12	12,53	87,47	89,95	34,89	97,83	82,92	82,76	82,94
13	33,42	66,58	87,38	76,24	92,98	85,89	87,95	84,86
14	49,21	50,79	84,16	81,47	86,76	84,20	83,45	84,92
15	53,31	46,69	86,28	82,11	89,94	86,16	88,16	83,88
16	9,27	90,73	92,01	19,97	99,37	84,10	68,78	85,66
17	1,72	98,28	98,30	1,14	100,00	96,28	42,01	97,22
18	0,60	99,40	99,40	0,00	100,00	98,86	13,73	99,37
19	1,06	98,94	98,94	0,00	100,00	97,70	27,68	98,46
20	2,68	97,32	97,33	0,15	100,00	93,83	46,71	95,12
21	9,82	90,18	91,60	23,07	99,07	82,65	69,79	84,05
22	59,01	40,99	86,21	92,71	76,85	86,13	88,37	83,05
23	47,39	52,61	84,33	86,08	82,76	84,31	88,68	80,37
24	2,12	97,88	97,99	5,17	100,00	96,12	45,76	97,22
25	0,16	99,84	99,84	0,00	100,00	99,86	37,5	99,96
26	0,00	100,00	100,00	0,00	100,00	100,00	0	100,00
27	0,05	99,95	99,95	0,00	100,00	99,94	0	99,99
28	0,25	99,75	99,75	0,00	100,00	99,49	6,35	99,72
29	6,19	93,81	94,49	17,22	99,59	88,89	64,43	90,40
30	45,98	54,02	82,99	82,96	83,01	82,99	87,76	78,93
31	60,96	39,04	85,93	92,47	75,72	85,55	88,07	81,61
32	10,79	89,21	92,94	41,77	99,13	89,52	76,46	91,09
33	0,08	99,92	99,92	0,00	100,00	99,92	19,05	99,99
34	0,00	100,00	100,00	0,00	100,00	100,00	0	100,00
35	0,00	100,00	100,00	0,00	100,00	100,00	0	100,00
36	0,05	99,95	99,95	0,00	100,00	99,91	28,57	99, 95
37	9,99	90,01	93,86	48,10	98,94	90,00	81,96	90,89
38	73,77	26,23	90,49	96,27	74,25	88,95	90,27	85,24
39	82,48	17,52	92,48	97,33	69,65	89,50	90,61	84,25
40	26,99	73,01	87,74	66,50	99,59	85,30	82,84	86,21
41	0,61	99,39	99,39	0,00	100,00	99,02	23,08	99,48
42	0,01	99,99	99,99	0,00	100,00	99,98	0	99,99
43	0,00	100,00	100,00	0,00	100,00	99,99	0	99,99
44	2,86	97,14	97,36	19,20	99,66	94,64	62,14	95,60
45	39,38	60,62	84,72	73,75	91,10	93,30	82,94	83,53
46	92,53	7,47	95,01	98,34	53,70	91,12	92,00	80,18
47	92,49	7,51	93,25	99,68	13,99	88,89	90,94	63,60
48	41,40	58,60	80,99	70,91	88,12	80,17	78,74	81,18
49	4,64	95,36	95,92	21,30	99,55	90,61	68,80	91, 67
50	0,18	99,82	99,82	0,00	100,00	99,53	0	99,71
51	1,22	98,78	98,77	1,60	99,98	96,69	47,12	97,30
52	8,66	91,34	93,89	44,77	98,55	87,78	81,35	88,38
53	73,85	26,15	85,73	94,47	61,04	82,96	84,47	78,72
54	97,64	2,36	97,66	100,00	0,83	95,15	96,46	40,93
55	92,73	7,27	93,27	99,85	9,33	87,52	90,50	49,51
56	87,08	12,92	88,37	99,27	14,90	80,48	83,04	63,26
57	10,62	89,38	92,21	31,67	99,40	85,24	76,97	86,22
58	1,34	98,66	98,66	0,00	100,00	95,41	45,48	96,09
59	4,98	95,02	96,38	32,36	99,74	91,93	80,63	92,52
60	25,26	74,74	83,95	51,66	94,86	79,42	79,42	79,42
61	89,64	10,36	90,71	99,59	13,82	82,30	84,58	62,57
62	89,87	10,13	90,82	99,60	12,88	81,42	83,24	65,26
Overall			*92,90*			*89,87*		

Table 1: SVM Training Results

training. A more thorough parameter search leading to a separate pair of parameters for every EPG point would certainly be preferable. Nevertheless, such a search would be far too much time consuming.

A spatial metaphor for time was used to model the dynamic nature of the speech signal, since there was no clear indication in the literature of an alternative existing in the SVM framework. We used a relatively small context window, of five time-frames, compared to other implementations in the acoustic-to-articulatory inversion field (eg. [10]). Selection of a larger window would be rather prohibitive in terms of training time needed, even though we use one of the fastest SVM packages available. The bulk of our data (originally almost 200.000 training examples – reduced to almost 40.000) stands near the limits of the capabilities of such packages. It is clearly problematic to use only subsets of our TIMIT data, since we would like our classifiers to capture all the connected speech processes in English.

Finally, for the purposes of this work we assumed independence among adjacent EPG events in space and time. Modelling such dependencies is a difficult machine learning problem. The Support Vector Machine framework, still in a relatively early stage, does not yet provide clear solutions to such problems. A few ideas seem to have appear in the literature (e.g. [14]), but nothing really concrete yet.

References

[1] Mike Brooks. The VOICEBOX toolkit. Available at http://www.ee.ic.ac.uk/hp/staff /dmb/voicebox/voicebox.html.

[2] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] 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.

[4] Steven B. Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pages 65–74. Morgan Kaufmann Publishers Inc., 1990.

[5] William J. Hardcastle, Fiona E. Gibbon, and Wilf Jones. Visual display of tongue-palate contact: Electropalatography in the assessment and remediation of speech disorders. *British Journal of Disorders in Communication*, 26:41–74, 1991.

[6] Chi-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Available at http://www.csie.ntu.edu.tw /˜cjlin/papers/guide/guide.pdf.

[7] Katerina Nicolaidis. An electropalatographic study of Greek spontaneous speech. *Journal of the International Phonetic Association*, 31(1):67–85, 2001.

[8] Edgar Osuna, Robert Freund, and Federico Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, Massachusets Institute of Technology, 1997.

[9] Matt Richardson, Jeff Bilmes, and Chris Diorio. Hidden-articulator markov models for speech recognition. In *ISCA IRTW Conference on Automatic Speech Recognition*, Paris, France, 2000.

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

[11] Todd A. Stephenson, Hervé Bourlard, Samy Bengio, and Andrew C. Morris. Automatic speech recognition using dynamic bayesian networks with both acoustic and articulatory variables. In *International Conference on Spoken Language Processing ICSLP2000*, Bejing, China, 2000.

[12] Asterios Toutios and Konstantinos Margaritis. A rough guide to the acoustic-to-articulatory inversion of speech. In *6th Hellenic European Conference of Computer Mathematics and its Applications, HERCMA-2003*, Athens, Greece, September 2003.

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

[14] Emmanuel Vazquez and Eric Walter. Multi-output support vector regression. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, 2003.

[15] Alan Wrench. A multi-channel/multi-speaker articulatory database for continuous speech recognition research. In *Phonus, Research Report No.4*, Institute of Phonetics, University of Saarland, 2000.