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}@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. 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; -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. It has also been suggested that visual feedback from EPG might be used in the context of second language acquisition.

However, there are difficulties in acquiring EPG data. First, each artificial palate must be individually manufactured from dental moulds of the speaker.

M. Faundez-Zanuy et al. (Eds.): NOLISP 2005, LNAI 3817, pp. 186–195, 2005.

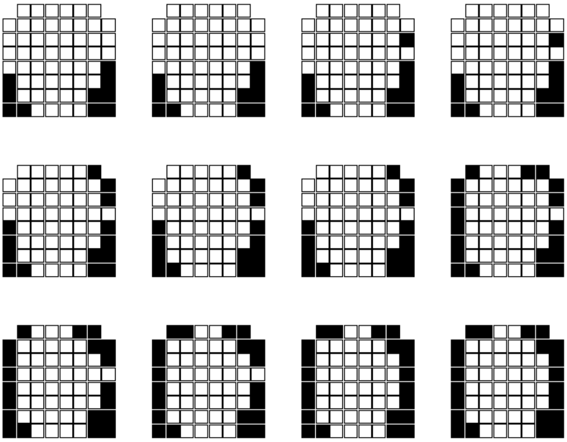

Fig. 1. Typical EPG sequence. Black squares indicate a contact between the tongue and the palate.

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 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 Overview of Machine Learning Techniques Used

3.1 C-Support Vector Classification

Given n training vectors $\mathbf{x_i}$ in two classes and a vector $y \in R^n$ 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 viewpoint) way [3, 4, 5] . The decision function that the C-SVC algorithm gives is:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} a_i y_i k(\mathbf{x}, \mathbf{x_i}) + b\right), \tag{1}$$

where b is a bias terms and the a coefficients are the solution of the quadratic programming problem:

$$\begin{gathered} \text{maximize } W(\mathbf{a}) = -\frac{1}{2}\sum_{ij} a_i a_j y_i y_j k(\mathbf{x_i x_j}) \\ \text{subject to } 0 \leq a_i \leq C, i = 1, \ldots, n, \text{ and } \sum_i a_i y_i = 0. \end{gathered} \tag{2}$$

Here C, called the *penalty parameter*, is a parameter defined by the user and $k(\mathbf{x_i x_j})$ 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:

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

where the γ parameter is selected by the user.

3.2 ϵ-Support Vector Regression

The ϵ-SVR algorithm [6, 5] generalizes the C-SVC algorithm 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 $y = f(\mathbf{x})$. According to ϵ-SVR, this estimate is:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (a_i^* - a_i) k(\mathbf{x_i}, \mathbf{x}) + b, \tag{4}$$

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

$$\begin{gathered} \text{maximize} \\ W(\mathbf{a}, \mathbf{a}^*) = -\epsilon \sum_{i=1}^{n} (a_i^* + a_i) + \sum_{i=1}^{n} (a_i^* - a_i) y_i - \frac{1}{2} \sum_{i,j=1}^{n} (a_i^* - a_i)(a_j^* - a_j) k(\mathbf{x_i x_j}) \\ \text{subject to } 0 \leq a_i, a_i^* \leq C, i = 1, \ldots, n, \text{ and } \sum_{i=1}^{n} (a_i^* - a_i) = 0. \end{gathered} \tag{5}$$

$C > 0$ and $\epsilon \geq 0$ are chosen by the user. C may be as high as infinity, while typical values for ϵ are 0.1 or 0.001.

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 data set 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

$$\mathbf{w_1} = \arg \max_{\|w\|=1} E\{\mathbf{w}^T\mathbf{x})^2\} \tag{6}$$

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:

$$\mathbf{w_k} = \arg \max_{\|w\|=1} E\left\{\mathbf{w}^T\left(\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w_i}\mathbf{w_i}^T\mathbf{x}\right)^2\right\}. \tag{7}$$

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 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, 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 *spatial metaphor for time*. That is, we construct input vectors spanning over a large number of acoustic frames. Based on some previous

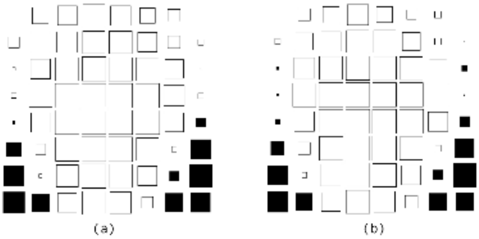

Fig. 2. Distributions of EPG events (a) in tthe training set (b) in the test. The bigger the square, the bigger the difference between positive and negative examples. Black squares indicate excess of positive examples and white squares excess of negative examples.

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.

Thus, we end up with training examples with a 221-dimensional (17 × 13) real-valued vector as input and a 62-dimensional binary vector as output. We split our data into two big halves: the even-numbered utterances constitute an "extended training set", and the odd-numbered ones an "extended test set". Each one has more than 100.000 examples.

But, since SVR training is a relatively slow process, using the whole "extended 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 phonemic label of each of our "extended training set" examples, we randomly select 200 training examples corresponding to every one of the 44 distinct phonemic labels. Since some phonemic labels have less than 200 examples in the dataset, we end up with 8686 training examples.

Finally, for our test set, we simply use 10 utterances spanning across our whole "extended test set". This test set consists of 5524 examples.

In both our final training and test sets, the distributions of the output among the EPG points values vary considerably, ranging from EPG points with a nearly equal number of positive (contacts, value 1) and negative (non-contacts, value -1) examples, to points with a 100% of examples belonging to one of the two classes. This fact is depicted graphically in Figure 2.

5 Training and Results

We follow two approaches to the mapping between the MFCCs and the EPG data. For the first one, 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.

The C-SVC algorithm then offers a straightforward way to independently deal with each one of these classification tasks, where the input is the MFCC vector

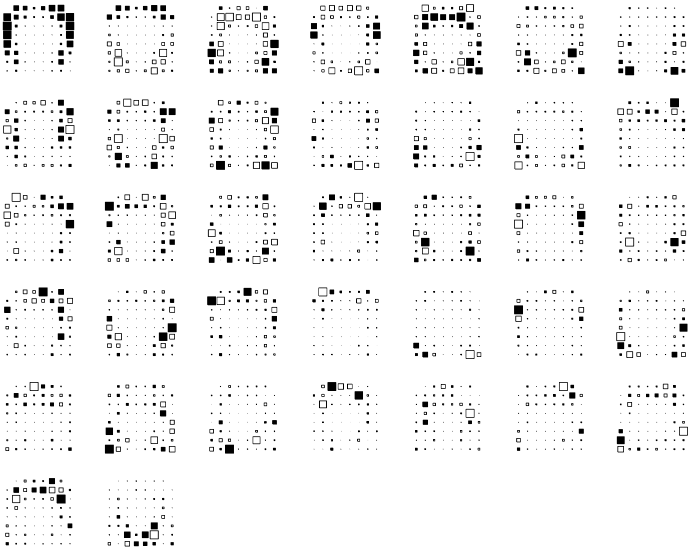

Fig. 3. Principal Components of the EPG data. Each value is represented by a square of size proportional to its absolute value and color black or white whether it is positive or negative.

(constructed as described previously) and the output is a binary value describing the activity of the EPG point in question.

We consider the RBF kernel with $\gamma = 0.0045$ and select $C = 1$, based on heuristics found in [10] The experiments are conducted using the LIBSVM software package [11].

For our second approach to the mapping, we consider accounting for the spatial relationships in the EPG data by applying PCA. We perform PCA on the "extended training set" and keep the 37 first principal components (depicted in Figure 3), which are the ones with eigenvalues larger than the 1/100 of the largest eigenvalue.

PCA transforms the output data by moving them into a new space. In this space the output values are real, so we have to solve 37 regression problems. We use ϵ-SVR for this task.

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

For the actual ϵ-SVR training, we use the RBF kernel with $\gamma = 0.0045$ and select $C = 1$ and $\epsilon = 0.1$. In testing, we need to invert the processes of scaling, centering and PCA.

For assessing the performance of are classifiers (even though we used regression in our second approach, the final outcome is still a set of classifiers) we

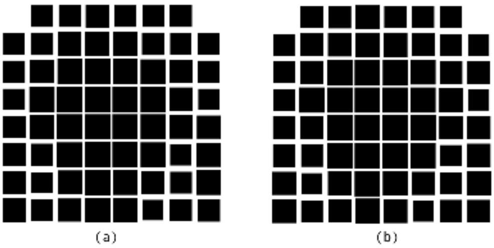

Fig. 4. Classification Rates for (a) the SVC approach (b) the PCA+SVR approach

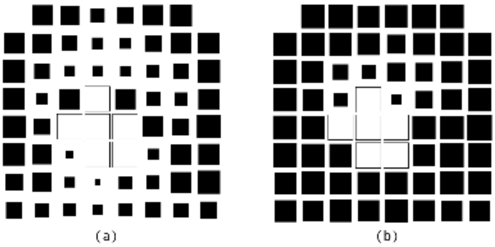

Fig. 5. AUCs for (a) the SVC approach (b) the PCA+SVR approach

use two metrics. The first one is the absolute classification rate (in the second approach by assigning positive output values as contacts, and negative as non-contacts), and the second one is the area under the ROC curve (AUC) [12]. The results are presented in Table 1 and Figures 4, 5. The convention used in the figures is that the size of the black squares is proportional to the value of the metric, while white squares indicate EPG points where the specific metric is meaningless (i.e. there is no AUC when all the examples in the test set belong the same class).

6 Conclusion

We applied two methods to the acoustic-to-electropalatographic mapping task, the first of which (SVC) does not take into account the spatial interrelationships inherent in the EPG data, while the second one (PCA+SVR) does.

The chance level (defined as the average percentage of the class with the most examples among the EPG points) of the data in the test set we used was 85,60%. Both the methods we applied exceed by far this chance level. For the SVC approach the average classification rate is 92,34%, and for the PCA+SVR approach 92,44%.

Between the two approaches, the differences in performance in terms of classification rates is small. The PCA+SVR approach improves upon SVC's classification rate only by 0,1%. Nevertheless, the ROC curves (with the exception of a couple of EPG points) are in general much better for the PCA+SVR, leading to

Table 1. Performances of the sets of classifiers in terms of Classification Rates and AUCs. Also shown the percentages of contacts in the training and test sets.

	Training Set	Test Set	SVC		PCA+SVR	
EPG Point	% Contacts	% Contacts	Class. Rate	AUC	Class. Rate	AUC
1	16,39	25,53	86,50	0,80	87,56	0,93
2	9,73	16,56	87,65	0,72	88,49	0,93
3	4,02	6,95	93,21	0,54	93,72	0,85
4	8,85	17,20	85,16	0,61	86,12	0,88
5	18,26	30,90	85,63	0,80	86,73	0,94
6	24,41	36,75	86,77	0,85	86,93	0,93
7	26,80	38,78	83,87	0,82	83,74	0,92
8	12,66	15,06	88,38	0,74	88,07	0,88
9	7,52	8,64	92,32	0,59	91,96	0,88
10	3,64	3,86	96,20	0,44	96,18	0,86
11	3,48	4,38	95,49	0,43	95,49	0,74
12	10,17	12,51	89,34	0,64	89,68	0,84
13	24,56	36,35	83,73	0,81	83,80	0,93
14	38,61	49,64	84,29	0,84	83,69	0,92
15	44,02	55,70	87,13	0,87	86,55	0,94
16	9,46	7,46	93,25	0,60	93,12	0,84
17	1,54	1,41	98,57	0,43	98,53	0,59
18	0,46	0,52	99,48	0,41	99,44	0,54
19	0,93	1,19	98,81	0,39	98,75	0,72
20	2,75	3,01	96,92	0,51	96,90	0,70
21	10,98	12,65	89,68	0,59	89,43	0,81
22	50,43	61,62	87,38	0,88	87,64	0,94
23	40,24	47,07	84,43	0,84	84,76	0,92
24	2,60	1,18	98,82	0,43	98,82	0,83
25	0,22	0,24	99,76	0,80	99,76	0,53
26	0,01	0,00	100,00	-	100,00	-
27	0,06	0,18	99,82	0,80	99,82	0,39
28	0,56	0,49	99,51	0,56	99,51	0,69
29	7,03	6,97	93,54	0,51	93,10	0,80
30	39,17	48,21	82,35	0,81	81,88	0,91
31	54,93	59,92	88,00	0,88	88,49	0,95
32	10,47	8,85	93,54	0,62	93,28	0,89
33	0,10	0,00	100,00	-	100,00	-
34	0,00	0,00	100,00	-	100,00	-
35	0,00	0,00	100,00	-	100,00	-
36	0,16	0,18	99,82	0,80	99,82	0,91
37	10,76	9,41	92,98	0,66	92,99	0,92
38	68,77	71,54	91,75	0,92	91,60	0,97
39	79,33	77,34	91,02	0,87	90,41	0,94
40	28,63	24,80	86,01	0,75	85,97	0,91
41	0,67	0,67	99,33	0,29	99,33	0,90
42	0,01	0,00	100,00	-	100,00	-
43	0,00	0,00	100,00	-	100,00	-
44	3,50	2,41	97,52	0,43	97,59	0,86
45	39,90	37,74	84,12	0,80	85,03	0,93
46	90,24	88,90	94,21	0,82	93,72	0,94
47	92,44	90,39	91,67	0,67	91,71	0,85
48	43,33	39,66	80,52	0,76	81,77	0,90
49	4,82	5,38	94,73	0,40	94,73	0,86
50	0,25	0,18	99,82	0,21	99,82	0,88
51	1,66	1,67	98,33	0,49	98,37	0,88
52	9,84	9,49	90,80	0,51	91,13	0,85
53	69,12	69,37	84,90	0,82	84,79	0,92
54	97,54	98,21	98,21	0,90	97,18	0,90
55	93,84	92,85	93,79	0,63	94,41	0,86
56	85,74	84,32	86,08	0,52	86,51	0,81
57	11,19	10,90	90,53	0,50	90,35	0,87
58	1,51	1,39	98,61	0,40	98,48	0,84
59	5,45	6,03	94,77	0,50	94,68	0,92
60	26,62	23,57	80,25	0,61	81,44	0,83
61	87,95	85,48	87,22	0,57	87,93	0,81
62	89,79	87,44	88,50	0,65	89,30	0,79
Overall			92,34	0,64	92,44	0,85

a remarkable increase in the average AUC, as shown in Table 1. Figure 6 shows the ROC curves for some characteristic EPG points.

So, it is mainly the improvement of the ROC curves achieved with the PCA+SVR approach, that makes it a better choice of an approach between the two. This agrees with the intuition that the PCA+SVR approach *should* be better, since it takes into account the spatial structure of the problem at hand.

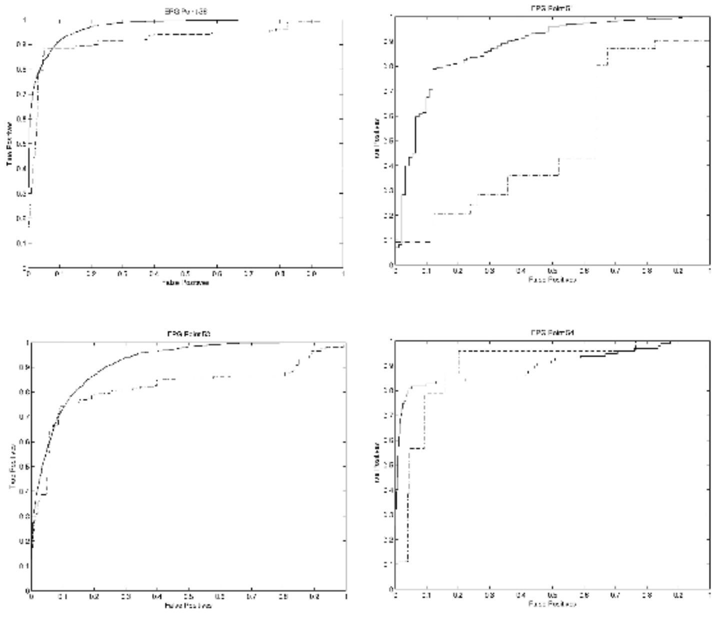

Fig. 6. ROC curves for some EPG points. Dashed-dotted curves correspond to the SVC approach, solid curves to the PCA+SVR approach.

One drawback of our experimental setup was that we trained our machines using only a small set of training examples, selected by a rather ad hoc procedure. As a future work direction, we might employ a more structured approach (i.e. clustering) in order to select training examples. Or, we might directly experiment with more data. Training time is always an issue, but recent findings in the machine learning field, such as Cross-Training [13], seem quite promising in the direction of speeding up things.

As a second future work direction, we could try to account for the temporal structure of our problem, i.e. the fact that the activity of a certain EPG point is depended on its activity at previous time instants. This is a difficult problem, though there are promising proposals from the machine learning field, such as the HMM–SVM method [14].

References

1. Miguel Á. Carreira-Perpiñán. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction.* PhD thesis, University of Sheffield, UK, February 2001.
2. 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.
3. Vladimir Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

4. Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
5. Bernhard Schölkopf and Alex Smola. *Learning with Kernels: Support Vector Machines, Optimization, Regularization and Beyond.* MIT Press, 1st edition, 2001.
6. Alex Smola and Bernhard Schölkhopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.
7. Mike E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.
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. Mike Brooks. The VOICEBOX toolkit. Software vailable at http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.
10. Jason Weston, Arthur Gretton, and Andre Elisseeff. SVM practical session (how to get good results without cheating). Machine Learning Summer School 2003, Tuebingen, Germany.
11. 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.
12. Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. Technical report, HP Laboratories, Palo Alto, April 2004.
13. Gökhan Bakir, Léon Bottou, and Jason Weston. Breaking SVM complexity with cross training. In *18th Annual Conference on Neural Information Processing Systems, NIPS-2004*, 2004.
14. Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. Hidden markov support vector machines. In *20th International Conference on Machine Learning ICML-2004*, Washington DC, 2003.