In: Proc. Int. Conference Signal Processing, Pattern Recognition, & Applications, IASTED, Rhodes, 2003, 98-103

Using Hidden-Markov-Models to analyze Concentrations from Biosensor Curves

Jörg Weitzenberg*
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg
D-06099 Halle (Saale), Germany
email: weitzenb@informatik.uni-halle.de

Stefan Posch
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg
D-06099 Halle (Saale), Germany
email: posch@informatik.uni-halle.de

ABSTRACT
In this paper a Hidden-Markov-Model (HMM) is adapted to analyze curves from amperometric biosensors. The main challenge is to detect a specific time interval during sampling, representative for the concentration, and to detect a time-point at which measurement may be terminated. This is in stark contrast to other applications of HMMs where they are implemented for pattern classification tasks. An appropriate algorithm for analysis is devised and a specific initialization of the Baum-Welch algorithm developed with a modification of the training algorithm itself. Results for a representative set of signal curve from different analytes are given.

KEY WORDS
Applications, Signal processing, Hidden-Markov-Models, Biosensors

1 Introduction

Hidden-Markov-Models are employed almost exclusively as the basic technique for automatic speech recognition systems since the late 80's. Other applications of HMMs have emerged in the course of the last years, for example face- and handwriting recognition [1, 2, 3] or analysis of biological sequences [4]. In this work we propose to apply HMMs for analysis of signal curves originating from amperometric biosensors. The aim of this analysis is to measure the concentration of analytes, like glucose, lactate, alcohols or hydrogen peroxide, in a given solution. A specific time interval within the signal has to be determined rather than recognition (or classification) of the signal which is in contrast to other applications of HMMs like speech or handwriting. Another feature, perhaps unique to the use of biosensors, is that a time point to terminate measurement must also be determined by the HMM, since it is not given a prior. These special requirements have been taken into account with appropriate modifications of the analysis and training procedures.

A simple method for analyzing biosensor curves has been proposed [5]. This method is based on a fuzzy logic pattern recognition system, needs only sparse resources of memory and computing time and therefore permits a straightforward implementation in microcontroller-based hand-held measuring devices. This method however employs only local processing of the curves and is unable to detect a suitable time point to terminate the sampling procedure. In this paper we describe an enhanced adaptive method using Hidden-Markov-Models. A first approach based on a set of discrete HMMs, one for each class of signal curves, has been published in [6], preliminary results of the more compact approach with one single HMM used in this paper have been described in [7].

In the next section we give a short overview of amperometric biosensors and the resulting signal curves. Our analysis on the basis of appropriate preprocessing using one single HMM modeling the different phases of the measurement and taking signal variations into account is described in section 3. Section 4 focuses on training of the resulting HMM, especially initialization of parameters and a modification of the Baum-Welch-Algorithm used. Results comparing different variants are presented in section 5, while section 6 gives some concluding remarks.

2 Amperometric biosensors

Biosensors are composed of a biological receptor component (e.g. an enzyme) and a transducer (see [8]). The analyte fits into the biological component of the sensor in analogy to a lock and accompanying key. As a result the physico-chemical properties of the component change and consequently the signal detected by the transducer changes as well, which is an electrical current $I(t, C_A)$ in our case (see figure 1). After an adjustment phase which typically includes an overshoot, the signal settles to a smooth progression in the main phase. However, the signal does not converge to a stable state, i.e. a constant current despite the smoothness of progression. Rather, there is only a relatively short time interval within this main phase where the current $I(t, C_A)$ takes the correct measurement value I_M to be detected, from which the analyte concentration C_A is to be deduced. A main aim of the analysis is to detect this particular time interval, called *measurement interval* in the following. As will be obvious form the discussion and

[*] Supported by SensLab GmbH Leipzig, Germany, http://www.senslab.de

examples in the next paragraphs, this measurement interval can only be detected using also a context of the curve subsequent to the measurement interval itself. Therefore a further requirement of the analysis is to determine a time point at which the sampling of the current may safely be terminated.

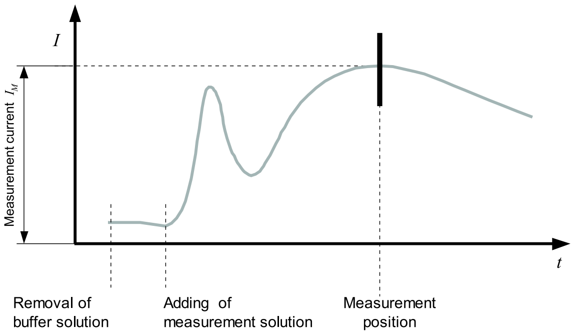
I
Measurement current I_M
t
Removal of buffer solution
Adding of measurement solution
Measurement position

Figure 1. Schematic shape of a signal curve showing the various events and phases.

The electrical current has linear characteristics for the measurement interval and over a sufficient range of concentration:

$$I_M = I_0 + I_D = I_0 + \alpha \cdot C_A,$$

where I_0 is a constant offset and I_D a concentration-depending signal-current. Thus, the scaling factor α has to be calibrated first. Subsequently, the currents I_0 and I_M have to be determined separately for each measurement in order to deduce the correct concentration.

In this work we concentrate on a specific procedure for single measurements with reusable amperometric biosensors fabricated by SensLab GmbH Leipzig (see [9]). The sensor has to be dipped into a buffer solution to determine I_0 first. The buffer solution is then removed from the sensor, which is called the preparation phase. Subsequently the sensor is dipped into the sample solution under consideration, resulting in a characteristic signal-curve composed of the adjustment and main phase introduced above. The shapes of theses curves vary considerably depending on analyte, sensor type, sensor handling, and concentration of the sample solution. In addition, also the measure interval differs severely between curves as figure 2 shows for different sensor handling and different concentrations. These variations are to be tolerated by the method presented in this article. As mentioned, the scaling factor α has to be calibrated in advance, which is accomplished with the same procedure using a solution of defined concentration.

3 Modeling and analysis

In this section our approach to model and analyze biosensor curves using HMMs is described, while learning of this HMM is deferred to section 4.

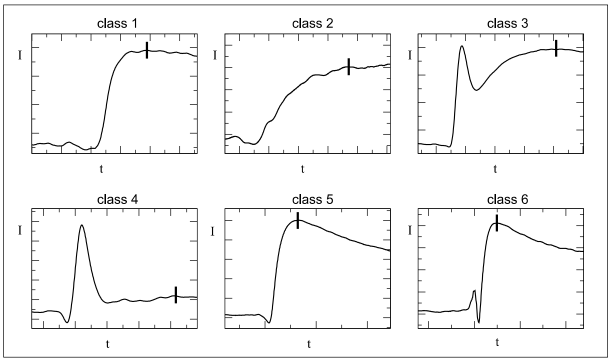

Figure 2. Typical curves for different sensor handling and low (class 3,4) resp. high (class 1,2,5,6) analyte concentrations. The point of time for correct measurement within the measure interval is marked by a small vertical line.

3.1 Preprocessing and feature extraction

The absolute sample values do not contain any relevant information to match the values measured to the various phases described in the last section and to determine the correct measurement interval. Therefore, an appropriate feature extraction mechanism has to be employed: The sampled values are first smoothed using a sliding average (window size of 30 samples) to reduce noise. The local characteristics are subsequently captured computing the first and second derivates of the sensor curve. Again a window of 30 samples is used to approximate the curve using a quadratic function and calculate its derivatives.

In case a discrete HMM is employed, these continuous features have to be quantized in addition. We use a two dimensional Self Organizing Map (SOM) [10] with 4×4 units in our system. Initial experiments yielded no improvement using a larger set of symbols.

3.2 Modeling of the measurement process

The resulting sequence of features, discrete or continuous, is now interpreted as originating from a discrete or semi-continuous HMM. The basic idea is to model the different phases of the measurement procedure, as described in the last section, with corresponding phases of a HMM (see figure 3). Within the HMM each phase is represented by one or more states using strict left-to-right state transitions. A feasible approach to cope with the different classes of curves is to model each class with its own HMM, as proposed in [6].

Since we are not interested in classification according to curve type, in this paper we use a single HMM instead. This results in a more compact solution better suited for an implementation in a hand-held measurement device. The main idea is to combine similar curve phases from different classes and represent each combined phase by a sequence of HMM states. This is motivated by the observation, that different curve classes as shown in figure 2, share a common shape of some phases. For example, the ad-

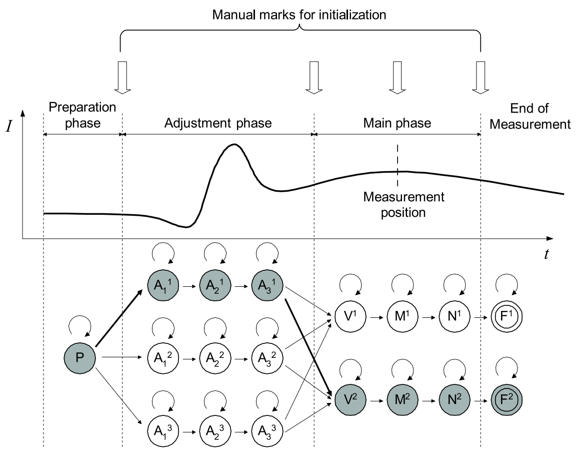

Figure 3. A single HMM (lower part) modeling a sample curve (middle part) is shown. The upper part shows manual marks as employed for training.

justment phase of class 3 and 4 is virtually identical, while on the other hand the main phase is equivalent for classes 5 and 6 respectively. These considerations result in the model shown in figure 3: The preparation phase (removal of buffer solution) is shared by all curves and modeled by a single state P. The observation sequence is subsequently interpreted to originate from one of m different parallel paths $\{A_1^1 \dots A_n^1\}$, $\{A_1^2 \dots A_n^2\} \ldots, \{A_1^m \dots A_n^m\}$ modeling the adjustment phase. In the HMM used for the results presented in this paper, we use three parallel paths of three states each, as depicted in figure 3.

The following main phase and *End of measurement* is again modeled with parallel paths, in this case k paths $\{V^1M^1N^1F^1\}, \dots, \{V^kM^kN^kF^k\}$. This main phase is differentiated into three sub phases: The measurement interval represented by states M^i is flanked by a pre- and post-measurement interval V^i and N^i respectively. The measurement value I_M is determined by maximization over all current values $I(t)$ within sub phase M^i. The shape and position of this main phase depends critically on the class of the curve under consideration. We use two parallel paths in our experiments (see figure 3) corresponding to the main phases of class 1 up to 4, and classes 5 and 6 respectively.

It is important to note that the (sub) phases N^i and F^i do not model differences in the underlying physical process with corresponding differences in the shape of the curve, but rather describe semantic discrimination: As soon as the HMM has stayed sufficiently long in the post-measurement phase N^i and reaches the state F^i we have adequate confidence to already have observed the measurement interval of the measurement and hence may end the measurement process.

3.3 Analysis

To allow for this detection of the end of measurement, The analysis proceeds for incrementally growing prefixes $\vec{o}$ of a virtually infinite sequence of feature vectors (see also figure 4 for pseudo code of the algorithm) to allow for the detection of the end of measurement. For each prefix $\vec{o}$ the optimal path $\vec{q}^*$ is computed using the Viterbi algorithm, i.e. the sequence of states which produces the observed prefix with maximal probability given the HMM λ. In case the last state q_T^* of this optimal path is one of the states F^i modeling the phase *End of measurement* the measurement interval to take the measurement value I_M as already been observed and the sampling of values $I(t)$ can be terminated. A maximum admissible time $T_{\max}$ is specified to take potential errors into account and the sampling of measurement values is terminated if exceeded.

```
$T := 0$
repeat
    $T := T + 1$
    fetch new sample value $I(T)$
    compute $o_T$ and derive new prefix $\vec{o} = (o_1, \dots, o_T)$
    compute optimal path $\vec{q}^* = \underset{\vec{q}}{\operatorname{argmax}} P(\vec{q}, \vec{o} \mid \lambda)$
                                        (Viterbi algorithm)
until $\left(q_T^* \in \{F^1, F^2 \dots, F^k\}\right) \vee (T = T_{\max})$

if $(q_T^* \in \{F^1, F^2, \dots, F^k\}) \vee$
   $\left(\exists t \leq T : q_t^* \in \{M^1, M^2, \dots, M^k\}\right)$
then
    $I_M = \max_t : q_t^* \in \{M^1, M^2, \dots, M^k\} I(t)$
else
    error
```

Figure 4. Pseudocode of the analysis

Using the determined optimal state sequence all current values $I(t)$ associated with one of the measurement states M^i are finally utilized to calculate the measurement value I_M. If the maximum specified time $T_{\max}$ has been exceeded and a final state F^i not been reached, however a measurement state M^i has already been reached within $\vec{q}^*$ at least once, it is still possible to determine a valid I_M. In this case a warning is given (which is not reflected in the pseudo code). If even this fails, the measurement has to be rejected with an error.

4 Training

An important property of HMMs is the availability of efficient procedures to learn the transition and emission probabilities characterizing the model from a set of sample curves. We use the well known iterative Baum-Welch algorithm from [11]. In case of a semi-continuous HMM we use the variant of Baum-Welch as described in [12], where the codebook is trained using feedback from the HMM estimation. In the following subsections we describe the initialization of the parameters and then two modification to the Baum-Welch algorithm specific to our application.

4.1 Initialization of parameters

Due to the properties of the Baum-Welch algorithm, the choice of suitable initial parameters – especially for the emission probabilities – is very important and are therefore estimated from the training curves. It is crucial for our application that not only the model is adapted to the curve as a whole, but in addition each states has to adopt as precisely as possible to the particular phase of the signal curve it models. Therefore we manually mark relevant positions within each curve of the training set to allow for a finer initialization of emission probabilities (see also figure 3). Besides the beginning and end of the measurement which are available anyway, only four relevant positions have to be marked using expert knowledge for each training sample to derive the duration of all phases and sub phases: The start of the adjustment phase, start and end of the main phase, and the precise point of time to take the measurement value I_M. From the last mark the sub phases corresponding to the measurement phase (modeled by state M^i) are derived as a symmetric intervals of 8 seconds duration. The initialization is treated differently for the preparation, adjustment, and main phase[1]. The emission probabilities of the preparation phase modeled by state P are initialized with the relative emission frequencies derived from the marked sections in all curves of the training set. Likewise, the emission probabilities for all states A^i_l in the m different parallel paths are first initialized identically from relative frequencies in the marked adjustment phase of sample curves. Subsequently random noise is added to the emission probabilities of these states to allow adaptation to different types of adjustment phases in an un-supervised way. In contrast, for the initialization of states in the main phase the manual classification of sample curves to curve classes is exploited: For each main phase $\{V^iM^iN^i\}$ the emission probabilities are initialized using only curves form the training set of the type assigned to this path. (As stated in subsection 3.2, curve class 1 up to 4, and classes 5 and 6 are assign to one of the parallel paths each.) Since also sub phases have been marked, emission probabilities for theses states can be initialized very precisely from the corresponding frequencies to allow for good localization of the measurement interval during analysis.

4.2 Modification of Baum-Welch algorithm

The additional knowledge supplied by the hand labeled intervals for the various phases in all sample curves may also be exploited to enhance the training procedure itself. For the standard Baum-Welch, in the maximization step of each iteration the transition and emission probabilities are updated. In short, this is accomplished using appropriate expectations with regard to the old HMM and the whole observation sequence $\vec{o}$. For example, the rule to update the the probability to emit Symbol O_k from state S_i to the new value $\hat{b}_{ik}$ is given by the standard Baum-Welch algorithm for the discrete case as follows:

$$\hat{b}_{ik} = \frac{\sum\limits_{t:o_t=O_k} \gamma_i(t)}{\sum\limits_{t=1}^{T} \gamma_i(t)},$$

where $\gamma_i(t) = P\left(q_t = S_i \mid \vec{o}, \lambda\right)$ is the probability to be in state S_i at time t with regard to the observation $\vec{o}$ and the old HMM λ of the last iteration.

Since hand labeling of phases constrain potential states of the HMM for a given symbol o_t of the observation, which is normally hidden to the outside, we are in the position to update only with respect to these *admissible* observations:

$$\hat{b}_{ik} = \frac{\sum\limits_{t:o_t=O_k \wedge S_i \text{ is admissible for } \mathrm{q_t}} \gamma_{ij}(t)}{\sum\limits_{t\in\{1,\ldots,T\} \wedge S_i \text{ is admissible for } \mathrm{q_t}} \gamma_i(t)},$$

A state S_i is said to be *admissible* for q_t of a given curve, if S_i is one of the states modeling the phase, o_t has been assigned to. For the sub phases of the main phase the curve class has to match as well. It is noted, that these are exactly the same constraints employed for the initialization as described in the last subsection. The update rule for the transition probabilities as well as update rules for the semi-continuous case are modified analogously.

4.3 Detection of end of measurement

As already stated, in addition to the detection of the measurement interval, an important problem is the detection of the end of measurement. As described in subsection 3.2, the interval of time required to reliably detect the end of measurement is modeled with states N^i and therefore with the transition probabilities from states N^i to F^i. The curve characteristics in the post-measurement phase, however, are identical for states N^i and F^i as already stated. Thus the transition probability cannot be estimated by the Baum-Welch algorithm. Therefore we train the HMM up to states N^i using complete observation sequences. Subsequently the final states F^i are added with the same emission probabilities of the corresponding state N^i. The state duration for each state N^i is estimated using the average duration of the post-measurement phase of the corresponding curve classes as previously marked in the training data. Using these average durations, the transition probabilities from states N^i to states F^i are calculated.

[1] No initialization is needed for the states F^i as discussed in subsection 4.3

5 Experimental results

In this section we give results for curves using reusable amperometric biosensors manufactured by SensLab GmbH. First we evaluate with regard to manually labeled measurement values I_M to compare different HMM and training variants while eliminating effects of inaccuracies of the sensor itself. Subsequently we compare against correct concentrations of the analyte, which is of course the real goal to achieve.

5.1 Accuracy of measurement values

The modified and unmodified Baum-Welch algorithm have been tested using a set of 405 glucose. The curves have been recorded using several glucose sensors of the same type at a sampling rate of 4 Hz. Different analyte concentrations and different types of sensor handling have been used to compile representative samples of curve shapes. In the case of the modified Baum-Welch, we additionally compared a discrete (DHMM) against a semi-continuous HMM (SCHMM) to test for its potential to improve the measurement results. A semi-continuous codebook with 16 Gaussian density functions was used for the SCHMM instead of the discrete codebook derived using the SOM.

The resulting mean relative errors of computed measurement value I_M using cross validation for evaluation with respect to the hand labeled value are given in table 1. Good accuracy is achieved for all variants and classes, while the modification of the Baum-Welch algorithm yields clear improvements. The more elaborate semi-continuous HMM did not however improve accuracy for this test set compared to a discrete HMM.

Class	unmodDHMM	modDHMM	modSCHMM
1	2.7 [%]	1.4 [%]	1.5 [%]
2	1.2 [%]	1.1 [%]	0.9 [%]
total	2.6 [%]	1.4 [%]	1.4 [%]

Table 1. Mean relative error of the computed measurement value I_M in respect to the hand labeled value for a set of 405 curves from glucose sensors: discrete HMM using unmodified Baum-Welch (unmodDHMM), modified Baum-Welch for a discrete HMM (modDHMM) and semi-continuous HMM (modSCHMM)

5.2 Accuracy of concentrations

Another series of experiments has been conducted to test the accuracy of concentrations computed. One single HMM was trained to analyze both glucose and lactate as analyte as a test for flexibility of the approach with respect to variations of curve shapes due to varying analytes. We used one individual glucose sensor, and two for lactate. As a training set, a total of 520 curves were compiled (241 glucose, 279 lactate), using 8 different concentrations (0.05mM, 0.1mM, 0.15mM, 0.2mM, 0.4mM, 0.6mM, 0.8mM, 1.0mM). To derive a separate test set, 11 experiments were performed (three for glucose and 8 for lactate as analyte), where in each experiments the same 8 concentrations were considered again. A sequence of 10 curves was sampled for each concentration and the scaling factor α (see section 2) was calibrated using a solution with standard concentration of 1.0mM for each set of 10 curves. The accuracy could thus be computed from a total of $11 \times 8 \times 10 = 880$ curves available to test the single trained HMM.

The modified Baum-Welch algorithm was used exclusively as it has shown to outperform the standard variant. The accuracies are given in table 2 supplemented with results using the older system based on fuzzy logic described in [5]. These computationally derived results are also compared to concentrations using the hand labeled measurement values I_M.

	mod DHMM	mod SCHMM	fuzzy	hand labeled
glucose	11.5 [%]	10.9 [%]	13.4 [%]	11.1 [%]
lactate	14.2 [%]	13.4 [%]	18.8 [%]	15.1 [%]

Table 2. Mean relative error for the computed concentration from the 880 curves of the independent test set: modified Baum-Welch for a discrete HMM (modDHMM) and semi-continuous HMM (modSCHMM), fuzzy logic system (fuzzy), and concentrations computed from hand labeled measurement value I_M (hand labeled)

The accuracy achieved is in the range of 10 to 15% for both HMM variants and the manually labeled. The fuzzy logic system is clearly inferior to the HMM based systems. The increase of relative error compared to accuracies of measurement values is due to the fact, that errors originating from the sensor itself are now also taken into account. Such errors could derive from a slight deviation from pure linear characteristics of the sensor for example. The semi-continuous HMM outperforms the discrete approach for this test set in contrast to the first set. The underlying cause for this effect lies in a larger variability of curves within this test set, due to a broader range of concentrations investigated and to the use two different analytes. The semi-continuous HMM is better suited to adapt to this variability since the lost of information due to vector quantization of continuous features is reduced decidedly. It comes as a surprise that the accuracy using hand labeled measurement values is seemingly inferior to the one achieved with the SCHMM. This is however probably caused by errors introduced during hand labeling. Incorrectly labeled measurement values I_M inevitably result in wrong concentrations derived. The trained HMM in contrast is able to generalize the subset of badly labeled examples in the training set and as a consequence outperforms manual evaluation.

Figure 5 shows the dependency of results with regard to different concentrations for one of the 11 experiments using glucose as analyte. As expected, the mean relative error decreases with increase of concentration due to increasing signal to noise ratio.

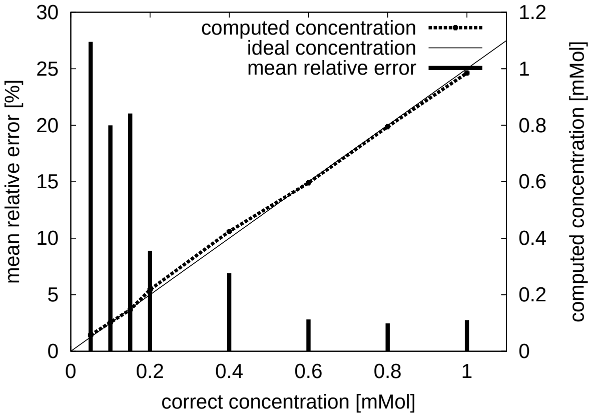

Figure 5. Dependency of the accuracy with respect to changing concentration. The figures given are for one of the three glucose experiments used to compile the test set, using the discrete HMM. The mean of the computed concentration for 10 curves is plotted against the ideal concentration. The mean relative error is shown as bars.

6 Conclusions

A method to analyze curves from amperometric biosensors using Hidden-Markov-Models has been presented. HMMs are successfully employed to model these signal curves taking variability of curve shapes into account. The standard approach of HMMs was modified in order to determine the relevant measurement interval within the signal curve and to reliably detect a time point to end measurement, which is not given a prior in contrast to most other applications of HMMs. A specific initialization of emission probabilities for the Baum-Welch algorithm and a modification of this training procedure itself have been proposed. The accuracies of detected measurement values and computed concentrations have been evaluated using almost 2000 signal curves. The application of a semi-continuous HMM improves results for a more demanding set of curves compared to the discrete HMM, while both HMMs yield comparable results for less demanding curves. All HMM based systems outperform a fuzzy based one due to its restricted local processing of signal curves.

To conclude, this approach based on HMMs results in a system ideally suited for a compact implementation even including hand-held measurement devices. It adapts automatically to various shapes of curves using a training procedure utilizing a hand labeled training set and yields accuracies comparable to if not better than manually determined values.

References

[1] Andrew W. Senior and J. Robinson, Anthony. An off-line cursive handwriting recognition system. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 20(3):309–321, 1998.

[2] Horst Bunke, Markus Roth, and Ernst Günther Schukat-Talamazzini. Off-line cursive handwriting recognition using Hidden Markov Models. *Pattern Recognition*, 28(9):1399–1413, 1995.

[3] S. Eickeler, S. Müller, and G. Rigoll. Gesichtserkennung mit Hidden Markov Modellen. In W. Förster, J.-M. Buhmann, and A. Faber, editors, *Mustererkennung '99, 21. DAGM-Symposium Bonn*, pages 348–355, Berlin, 1999. Springer-Verlag.

[4] Pierre F. Baldi and Søren Brunak. *Bioinformatics: The machine learning approach*. The MIT Press, 1998.

[5] J. Weitzenberg, S. Posch, C. Bauer, M. Rost, and B. Gründig. Analysis of amperometric biosensor data using fuzzy logic and discrete hidden-markov-models. In H.R. Tränkler, editor, *SENSOR 2001*, pages Volume II: 493–498, 2001.

[6] J. Weitzenberg, S. Posch, and M. Rost. Diskrete Hidden Markov Modelle zur Analyse von Meßkurven amperometrischer Biosensoren. In G. Sommer, N. Krüger, and Ch. Perwass, editors, *Mustererkennung 2000, 22. DAGM Symposium Kiel*, pages 317–324, Berlin, 2000. Springer Verlag.

[7] Jörg Weitzenberg, Stefan Posch, and Manfred Rost. Analysis of amperometric biosensor curves using hidden-markov-modells. In Luc Van Gool, editor, *Pattern Recognition. 24. DAGM-Symposium*, pages 182–189, 2002.

[8] E. A. Hall. *Biosensoren*. Springer-Verlag, Berlin/Heidelberg, 1995.

[9] Bernd Gründig. Amperometrische Enzymsensoren. In G. Henze, M. Köhler, and J.P. Lay, editors, *Umweltdiagnostik mit Mikrosystemen*, pages 257–303. Wiley VCH, 1999.

[10] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Heidelberg, 1995.

[11] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[12] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.