Biol. Cybern. 71, 359–373 (1994)

Biological Cybernetics

Analysis, classification, and coding of multielectrode spike trains with hidden Markov models

G. Radons [1], J. D. Becker [2], B. Dülfer [3], J. Krüger [4]

[1] Institut für Theoretische Physik, Universität Kiel, Olshausenstrasse 40, D-24118 Kiel, Germany
[2] Fakultät für Physik, Universität Freiburg, Hermann-Herder-Strasse 3, D-79104 Freiburg, Germany
[3] IMIT, Hahn-Schickard-Gesellschaft, Hahn-Schickard-Strasse 10, D-78054 Villingen-Schwenningen, Germany
[4] Neurologische Universitätsklinik, Hansastrasse 9, D-79104 Freiburg, Germany

Received: 31 August 1993/Accepted in revised form: 8 March 1994

Abstract. It is shown that hidden Markov models (HMMs) are a powerful tool in the analysis of multielectrode data. This is demonstrated for a 30-electrode measurement of neuronal spike activity in the monkey's visual cortex during the application of different visual stimuli. HMMs with optimized parameters code the information contained in the spatiotemporal discharge patterns as a probabilistic function of a Markov process and thus provide abstract dynamical models of the pattern-generating process. We compare HMMs obtained from vector-quantized data with models in which parametrized output processes such as multivariate Poisson or binomial distributions are assumed. In the latter cases the visual stimuli are recognized at rates of more than 90% from the neuronal spike patterns. An analysis of the models obtained reveals important aspects of the coding of information in the brain. For example, we identify relevant time scales and characterize the degree and nature of the spatiotemporal variations on these scales.

1 Introduction

The analysis of multielectrode data and the extraction of information about the coding principles in the brain are difficult tasks. This is mainly due to the following characteristics of the measured data: The observed processes are in general nonstationary, and they exhibit large variations, which usually cannot be explained by simple noise but may show systematic fluctuations that may have hidden meanings. In addition, the data are high-dimensional. For instance, we have to deal with 30 degrees of freedom corresponding to a measurement with 30 electrodes.

The experimental data treated in this work consist of simultaneous recordings of spike trains with 30 microelectrodes from the visual cortex of an anesthetized and paralyzed monkey (Krüger and Aiple 1988). We are interested in the neuronal responses recorded during the application of various visual stimuli. In this work, we investigate the responses to bars moving in different directions. This gives rise to corresponding classes of neuronal activity at the electrode array, which are in some sense characteristic for the applied stimuli. It turned out (Krüger and Becker 1991) that the 30 mean firing rates or the spike counts at each electrode during the relevant response intervals contain little information about the currently applied stimulus. Instead, this information was found in the recorded spatiotemporal discharge patterns. This is in accordance with results from other experiments (Richmond et al. 1987). Thus, our goal consisted in analyzing and characterizing such spatiotemporal excitation patterns.

As a first step in this direction, one has to evaluate, whether or not one can assign the corresponding visual stimulus to an observed spike pattern. Or, in other words, can one recognize the visual stimulus from the elicited neuronal discharges? It should be borne in mind that it is in this way that the animal makes use of neuronal excitations. This problem is basically a pattern recognition task which can be tackled with classical methods. For our data one can successfully apply linear classifiers (Krüger and Becker 1991) or non-linear classifiers such as artificial neural networks. A result of these investigations is that the *spatio-temporal* patterns contain relevant information about the applied stimulus. These methods, however, do not take into account that the patterns are generated dynamically. Furthermore, it is difficult to infer which properties of the patterns led to the discrimination, to what extent they are of a statistical nature, and what are the characteristics of the various stochastic components of the patterns.

Possible approaches for the solution of these questions could combine spectral methods like principle component analysis, as used, e.g., in Richmond et al. (1987), and classical pattern recognition techniques. Here

Correspondence to: G. Radons

we present results based on the use of hidden Markov models (HMMs) and corresponding parameter estimation techniques (Baum et al. 1970). These models, which are otherwise known as stochastic automata (Paz 1971) or probabilistic functions of Markov chains (Baum et al. 1970), have been very successfully applied in various speech recognition tasks (for reviews see, e.g., Rabiner 1989; Bahl et al. 1983; Huang et al. 1990). In these applications, the data, energies in the time-frequency domain, show a similar degree of complexity, nonstationarity, and large variability as in our case. This led us to investigate whether HMMs are equally well suited for the classification and analysis of our multielectrode data (Radons et al. 1992). The idea of modeling spike data with HMMs was independently developed by two other groups. In Pawelzik et al. (1993), the experimentally observed oscillatory neuronal responses in the cat's visual cortex (Gray et al. 1989) are explained in terms of an HMM, while in Gat and Tishby (1993), two behavioral modes of monkeys are identified with the aid of HMMs. In contrast, our work treats the problem of modeling and distinguishing the neuronal responses to many external stimuli, which implies that, similar to speech recognition problems, we have to work with an ensemble of different HMMs. Another biophysical problem in which HMMs were applied successfully is the analysis of ion currents through channels of cell membranes (Chung et al. 1990, 1991; Fredkin and Rice 1992; Becker et al. 1994).

The advantage of using HMMs for the analysis of multielectrode data is threefold: Beyond being a pattern recognition and classification tool, it provides us with probabilistic dynamical models of the pattern-generating process. This implies the possibility of reproducing the data with a reduced set of parameters, and thus serves as a data compression method; on the other hand, it preserves the possibility to extract, e.g., various correlation functions or correlograms. The third point is that the extracted models, although of abstract nature, are amenable to an analysis in terms of subprocesses, if present, which contribute to the pattern-generating process as a whole.

In Sect. 2 we briefly describe the nature of the data to be analyzed, and we introduce the principles of HMMs and the variants tested in this work. Section 3 is devoted to the presentation of results, where we compare the performances of the various models in terms of recognition rates and demonstrate the quality in reproducing the original data. In Sect. 4, the results are discussed. The Appendix consists of a collection of formulas used in the parameter estimation procedures.

2 General aspects

2.1 The data

The data were recorded with 30 microelecrodes in layer VI of the striate cortex of a paralyzed and anesthetized monkey. Electrodes were arranged in a 5×6 array with a spacing of 160 μm. Thus, the electrodes were located in an area of 0.64×0.8 mm^2. They were labelled A–E (columns) and 1–6 (rows). Figure 1 shows two typical recordings of the neuronal response to the same stimulus. Note the large variability of the spike patterns. The spikes were recorded at a sampling rate of 1 ms. For about half of the electrodes, the recorded signals stem from one cell. At the remaining electrodes, contributions from more cells cannot be excluded, although typically spikes from one cell are dominant. The stimuli were monocularly presented bright bars on a dark background. The bars were 7 min of arc wide and moved at 1 min of arc per 10 ms. One trial lasting 40 s consisted of a sequential presentation of the bar moving in sixteen equally spaced directions. The experiment was repeated 21 times, and therefore the data consist of 21 trials, i.e., repetitions of 16 stimuli. Responses to different stimuli are well separated in time. A more detailed description of the measurement can be found in Krüger and Aiple (1988).

2.2 Principles of HMMs and their application to multidimensional spike trains

In the following, we briefly survey the general principles of HMMs and their application to spike data. In Sect. 2.2.1 we describe how a given HMM is used as a probabilistic generator of symbol sequences. The meaning of the symbols varies from application to application and is described for spike data in Sect. 2.2.4. Section 2.2.2 reviews how an ensemble of HMMs may serve as a classifier or pattern recognition tool. Such an application presupposes optimized models which can be found by 'learning' procedures also described in this section. How such optimized HMMs can be analyzed is explained in Sect. 2.2.3.

2.2.1 HMMs as probabilistic dynamic models. An HMM is an abstract object consisting of a given number of states i, $i = 1, \ldots, N$ and transitions between these states. Transitions occur with probabilities a_{ij}, i.e., a_{ij} is the conditional probability $p(j \mid i)$ for making a transition to state j, if the system is in state i. The a_{ij} have the property $\sum_{j=1}^{N} a_{ij} = 1$ for all i, which means that some transition occurs with probability 1. Therefore, they can be considered as elements of a stochastic matrix A, the transition matrix.

So far, this defines a simple Markov process, because the probability for the next state j depends only on the current state i. HMMs are characterized by the additional ingredient that for every state i one defines a probability distribution $b_i(S)$ for emitting a symbol S of some alphabet $\{S\}$ of length $|S|$. The alphabet may also consist of infinitely many symbols $|S| = \infty$. Some symbol is generated with certainty in every state i, which means that $\sum_{\{S\}} b_i(S) = 1$ for every i. The $b_i(S)$ are, in general, different functions depending on i. The meaning of the symbols is application-dependent and is introduced for our problem in Sect. 2.2.4 below. The above definitions explain why HMMs are often called probabilistic functions of Markov processes.

In order to generate symbol sequences with such models, one also has to specify an initial probability distribution $\vec{\pi} = (\pi_1, \ldots, \pi_N)$ over the states i. A symbol

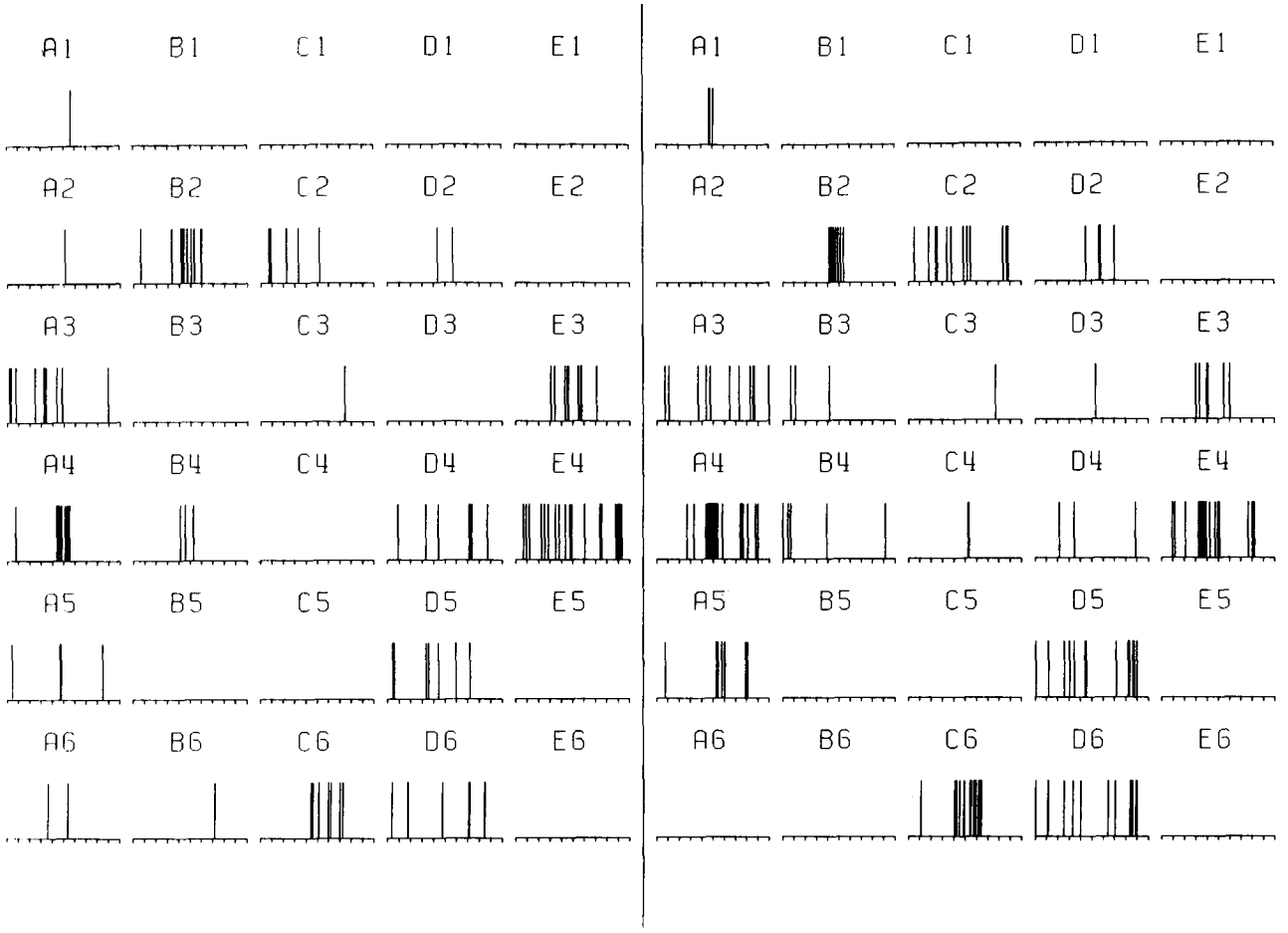

Fig. 1. Display of two different responses to the same oriented moving bar. The duration of the records shown is 800 ms. The electrodes were physically arranged in the same way as the records in this picture. Each *vertical line* represents a spike. Although both responses were recorded from the same neurons almost immediately after each other, the responses are quite different

sequence of length T, $S_1 S_2 \ldots S_T$ with $S_t \in \{S\}$, $t = 1, \ldots, T$ is generated as follows: One randomly selects an initial state according to the distribution $\vec{\pi}$, e.g., state i with probability π_i, and emits a symbol S_1 with probability $b_i(S_1)$; then one jumps to another state, say j with probability a_{ij}, and emits symbol S_2 with probability $b_j(S_2)$, and so on. The probability $p(S_1 \ldots S_T)$ for generating a sequence $S_1 \ldots S_T$ is obtained by summing the probabilities from all paths through the automaton which are compatible with this sequence. It can be calculated as

$$p(S_1 S_2 \ldots S_T) = \vec{\pi} B(S_1) A B(S_2) A \ldots A B(S_{T-1}) A B(S_T) \vec{\eta} \tag{1}$$

where we introduced the diagonal matrices $B(S)$, $S \in \{S\}$, with elements $b_{ij}(S) = b_j(S) \cdot \delta_{ij}$. The multiplication with the vector $\vec{\eta} = (1, 1, \ldots, 1)^T$ provides a summation over all states, which means that the system is allowed to be in any of the N states j while it emits the last symbol S_T. In the speech recognition literature, the evaluation of (1) is called a forward-backward algorithm, which reflects that the product can be calculated iteratively from left to right (forward) or from right to left (backward).

To summarize, an HMM is defined by a set of parameters which includes the initial probability distribution $\vec{\pi}$, the transition matrix A, and the symbol generating matrices $B(S)$ with S from the alphabet $\{S\}$. These parameters are conveniently collected in one vector denoted $\vec{\lambda}$, that is

$$\vec{\lambda} = (\vec{\pi}, A, \{B(S)\}) \tag{2}$$

In view of (1) and (2), one can identify a given HMM also with a parametrized probability distribution over all symbol sequences $S_1 S_2 \ldots S_T$ of arbitrary length T. We denote this probability distribution in the following by $P(S_1 S_2 \ldots S_T | \vec{\lambda})$.

The connectivity structure of an HMM, or its topology, tells which states are connected or which transitions between states are allowed regardless of the probabilities attached to them. Formally, this is described by the adjacency or connectivity matrix C with elements $c_{ij} = 1$, if $a_{ij} \neq 0$, or $c_{ij} = 0$ if $a_{ij} = 0$. This matrix is often represented diagrammatically by connecting states or nodes by arrows, if the corresponding transitions are allowed. There are two important classes of topologies which are referred to as ergodic models and as left-right

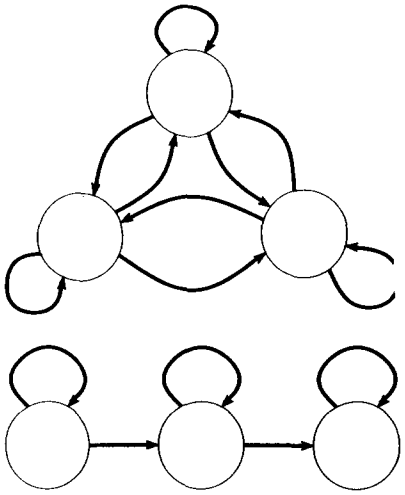

Fig. 2. Examples of the two basic topologies used in this work: **a** a fully ergodic model (*top*), **b** a strict left-right model (*bottom*)

models. A three-state model of each class is depicted in Fig. 2.

Ergodic models are characterized by the fact that each state can be reached from each other state with some sequence of transitions. In Sect. 3 we present results for special ergodic models in which none of the possible transitions is forbidden, which means that the connectivity matrix contains no zeros (corresponding to Fig. 2a). Left-right models have the property that their connectivity matrix is an upper triangular matrix, i.e., it contains zeros everywhere below the diagonal, which implies that eventually all transitions end in an absorbing final state. Our results below were obtained for restricted left-right models characterized by a connectivity matrix with ones in the diagonal and the first upper off-diagonal, and with zeros elsewhere (see Fig. 2b). In the following, these two types of models are for simplicity referred to as ergodic and left-right models, respectively, although they are actually special cases of these classes. The importance of the topology of an HMM lies in the fact that for the 'learning' processes described in the next section one has to pre-assume one of the possible topologies which is preserved during the learning procedure, and this may have consequences for the resulting properties of the model obtained. It should be mentioned that methods exist for minimizing previously obtained models and for checking the equivalence of given HMMs (Paz 1971). Thus, the extraction of a large enough ergodic model and subsequent minimization should lead to a unique model or to an equivalence class of minimal models. For finite data sets, however, these methods are currently not available.

2.2.2 HMMs as classifiers. In a classification task, one has a given number of classes of patterns, and one wants to decide for some pattern to which class it belongs. This is often done by calculating the distance (within some metric) of the given pattern to some representative of each class (Duda and Hart 1973). Then one decides that it

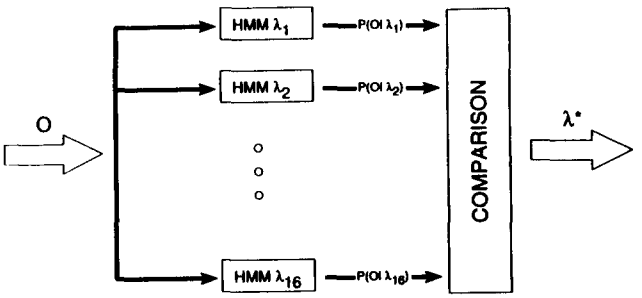

Fig. 3. Scheme of the classification process with hidden Markov models (HMMs): An incoming signal O is attributed to the HMM which generates this signal with the highest probability

belongs to the class where the distance is minimal, if this distance is not too large.

HMMs can be used for pattern classification as follows: For each class of, say, K classes of patterns, one has to design one HMM, each of which is a representative of a different type of pattern. In our application $K = 16$, corresponding to the 16 different visual stimuli which produce 16 classes of neuronal responses. In order to use HMMs for pattern classification, one has to translate the pattern into a symbol string, which may also be considered as a possible output of the given HMMs. We call this symbol string, which corresponds to some input pattern to be classified, the observation sequence $O = S_1 S_2 \ldots S_T$. Now one simply decides that O belongs to that class whose associated representative HMM produces O with the highest probability. This probability serves as a discriminant function for the classification process (Duda and Hart 1973). Schematically, this is depicted in Fig. 3.

Thus, one has to calculate for each HMM $\vec{\lambda}_k$, $k = 1, \ldots, K$ the probability $P(O|\vec{\lambda}_k)$ that model $\vec{\lambda}_k$ produces the observation sequence O. This is done with the forward-backward algorithm (1). The model $\vec{\lambda}^*$ for which $P(O|\vec{\lambda})$ is maximal is regarded as the representative of the correct class, formally

$$\vec{\lambda}^* = \text{argmax}_{\vec{\lambda}_i} P(O|\vec{\lambda}_i) \tag{3}$$

This comparison can in principle be done with an arbitrary ensemble of K HMMs, and one would always get an answer, i.e., a classification of the input pattern. In order to make sense, the classification process has to be carried out with an ensemble in which the HMMs really represent in some sense the corresponding pattern classes. This is achieved by a training or learning procedure in which the parameters of each HMM are optimized in the following sense: If for each pattern class one had only one given representative symbol sequence $O^{(k)}$, $k = 1, \ldots, K$ one would train each HMM such that model $\vec{\lambda}_k$ produces $O^{(k)}$ with the maximal probability. This is usually done by iteratively optimizing $\vec{\lambda}_k$, $\vec{\lambda}_k(t=0) \rightarrow \vec{\lambda}_k(t=1) \rightarrow \vec{\lambda}_k(t=2) \rightarrow \cdots \rightarrow \vec{\lambda}_k$, such that in each step the likelihood $P(O^{(k)}|\vec{\lambda}_k(t))$ of producing $O^{(k)}$ increases (or remains the same), i.e., $P(O^{(k)}|\vec{\lambda}_k(t=0)) \leqslant P(O^{(k)}|\vec{\lambda}_k(t=1) \leqslant \cdots \leqslant P(O^{(k)}|\vec{\lambda}_k)$. With such a learning procedure, one obtains K HMMs, each producing the representative sequence

$O^{(k)}$ with the maximal probability, and the above-described decision process for an incoming new observation sequence O makes sense.

In general, and also in our application, a class is not represented by only one string $O^{(k)}$ but by an ensemble of samples $\{O^{(k)}\}$ which are known to belong to class k. Since the members of $\{O^{(k)}\}$ may occur with different frequencies, the class k is actually characterized by a probability distribution $P(O^{(k)})$ over the samples. In this case, one wants the probability distribution $P(O^{(k)} \mid \vec{\lambda}_k)$ over sequences produced by model $\vec{\lambda}_k$ to be as similar as possible to $P(O^{(k)})$.

This can be achieved by iteratively optimizing $\vec{\lambda}_k$, such that the Kullback-Leibler distance D

$$D(P(O^{(k)}) \| P(O^{(k)} \mid \vec{\lambda}_k)) = \sum_{\{O_{(k)}\}} P(O^{(k)}) \log \frac{P(O^{(k)})}{P(O^{(k)} \mid \vec{\lambda}_k)} \qquad (4)$$

is minimized during the optimization procedure. One has $D \geqslant 0$ with equality only if $P(O^{(k)})$ and $P(O^{(k)} \mid \vec{\lambda}_k)$ are identical.

In principle, any of the known optimization algorithms such as gradient descent or stochastic methods like simulated annealing could be used for finding the optimal parameter vector $\vec{\lambda}_k$ which minimizes the Kullback-Leibler distance $D(\vec{\lambda}_k)$ (4) or maximizes the likelihood $P(O^{(k)} \mid \vec{\lambda}_k)$ for a given class k. The most widely used method for HMMs is the so-called Baum-Welch re-estimation algorithm (Baum et al. 1970), which is a variant of the expectation-maximization (EM) algorithm (Dempster et al. 1977) of mathematical statistics. It has the general form

$$\vec{\lambda}_k(t+1) = \vec{f}\,[\vec{\lambda}_k(t), \{O^{(k)}\}] \qquad (5)$$

where $\vec{f}$ is a relatively complicated function of the old parameters $\vec{\lambda}_k(t)$ and the observation samples. The explicit form of $\vec{f}$ is given in the Appendix for the two variants of the algorithm which were used in this work One should note that with any optimization algorithm there exists the possibility of becoming stuck in local minima of $D(\vec{\lambda}^{(k)})$ or maxima of $P(O \mid \vec{\lambda}^{(k)})$. To avoid this, one has to run the optimization algorithm under several different initial conditions $\vec{\lambda}^{(k)}(t=0)$.

A possible criterion for the quality of the obtained models is the rate of correct classifications of patterns or sequences not used in the training phase. Another measure could be, e.g., the degree of correct reproductions of correlation functions. For our data, the recognition rate appears to be the most valuable criterion. A good classification and discrimination ability is a necessary condition for further investigations of the properties of the inferred models.

2.2.3 Analyzing HMMs. Given a representative of a class of patterns in the form of an optimized HMM, one can ask what the most probable sequence or pattern generated by the model will be. This defines something like a prototype pattern of the class being considered. One should emphasize that this is in general different from some average over the pattern samples. It is rather a very 'pure' version or a typical pattern of the class. The situation is as for general probability distributions: For multimodal distributions, it may happen that the average value of a random variable is not realized by any of the samples, whereas the most probable sample is always meaningful. To obtain such a prototype sequence, one has to evaluate (1) for all possible sequences of given length T and compare the resulting probabilities. This method is feasible only for relatively short sequences ($T \approx 10^1 - 10^2$) since the number of strings increases exponentially with T. In our case, this method is still applicable. An alternative consists in searching for the most probable path of given length through the automaton. The associated symbol string is often also the most probable sequence. This problem can be solved by the Viterbi algorithm (Forney 1987), a simple linear programming technique with complexity increasing linearly in T. Note, however, that optimizing an HMM is similar to curve fitting, and therefore, it may happen that the HMM extrapolates into regions of pattern space where no samples exist for training the HMM. A maximum of the probability distribution in such a region is usually unreliable, and one has to restrict oneself to sequences in the range of the training samples. Note also that one needs some distance measure between patterns or symbol sequences in order to check which situation prevails.

A related topic is the question of whether other relative maxima also exist in the probability distribution over the sequence space. Such secondary maxima correspond to a situation in which a class of patterns is made of several subclasses which manifest themselves as distinct clusters in pattern space. Physically, this would mean that there is not only the feature present which defines the class, but also secondary features which may or may not be known beforehand. As above, one has to investigate neighborhoods of strings, and therefore one has to define some metric in symbol space which reflects neighborhood relations in pattern space. The advantage of using HMMs instead of looking at the pattern space directly is that one has a simplified representation of the data, which makes it easier to recognize such subfeatures and, e.g., to find the subclass to which a new unknown pattern belongs. A very simple way of representing such different subclasses in an extracted HMM is by different groups of paths. In our application, this may help to decide which processes or features correlate with others on different electrodes.

2.2.4 Application to spike data. The first problem one encounters in applications of HMMs to continuous time problems such as speech recognition and also spike data analysis is the segmentation problem, i.e., the dividing of the signal into time bins or segments of some length in order to obtain a discrete 'time' process described by HMMs as a jump process between the available states. The various possibilities we explored are described in Sect. 3.1. The second problem in using discrete HMMs consists in obtaining a symbolic or discrete representation of the data for each time segment. For spike data, a very natural alphabet exists: One has to assign to every time segment simply the number of spikes contained in it. For one electrode, the data are then represented as a

sequence of numbers, which vary between zero and the maximum number of spikes encountered in the segments. The problem lies in the fact that in our experiment we have 30 electrodes. If the number of spikes per segment and electrode varies for instance from 0 to 9, this would lead to an alphabet with 10^{30} symbols if every possible combination of spike counts on the electrodes is taken into account by a distinct symbol. As explained in Sects. 2.2.1 and 2.2.2, one has to optimize a probability distribution $b_j(S)$ on every state j of a given model. For $|S| = 10^{30}$, this is clearly impossible. There are basically two approaches to circumvent this problem. The first consists in a suitable coarse-graining of the alphabet to obtain a smaller set of symbols. This method is basically the same as vector-quantizing continuous variables (Gray 1984; Makhoul et al. 1985): One has to design a code book that tells which spatial firing pattern, i.e., the vector of spike counts at each electrode in some time segment, including some neighborhood is coded by what symbol. Inevitably, there is some information loss, and also the individuality of the single electrodes is abandoned and replaced by an overall pattern. The detailed methods used for the design of the code books and the results for the quality of vector quantization (VQ) are presented in Sect. 3.1. The HMMs based on this method are called VQ-HMMs in the following, and their performance is reported in Sect. 3.2 and compared with a second alternative. This alternative, in reducing the parameters which characterize the output probability distributions, lies in making assumptions about the functional form of $b_j(S)$. This amounts to parametrizing these functions and optimizing the corresponding parameters. It turns out that assuming a multivariate Poisson process or a binomial distribution on the nodes of the HMMs accounts well for the actually observed spike statistics in individual time segments. The re-estimation formulas for such output processes are derived in Becker (1994) and in Dülfer (1993). We can even assume independency of the spike distributions for each electrode on each node j. In the Poisson case, this means that we assume the following form of the output probabilities

$$b_j(\vec{k}) = \prod_{l=1}^{30} b_j^{(l)} = \prod_{l=1}^{30} P(k_l, \mu_j^{(l)}) \tag{6}$$

where $P(k, \mu)$ is a Poisson distribution with mean value μ

$$P(k, \mu) = \frac{1}{k!}\mu^k e^{-\mu} \tag{7}$$

Thus, $P(k_l, \mu_j^{(l)})$ is the probability of observing k_l spikes on electrode l in some time segment of length Δ, if the system is in state j. The corresponding mean number of spikes is $\mu_j^{(l)}$. For binomial output processes, $P(k_l, \mu_j^{(l)})$ is replaced by $B_n(k_l, \mu_j^{(l)})$ where $B_n(k, \mu)$ is the binomial distribution

$$B_n(k, \mu) = \binom{n}{k}\mu^k(1 - \mu)^{n-k} \tag{8}$$

The additional parameter n is the maximum number of spikes per time interval, which we keep fixed in the optimization process. The only free parameter which is adapted is μ or the mean spike number $n \cdot \mu$.

Thus, in both cases, we have to optimize only 30 parameters for every state j instead of 10^{30} as in the original formulation. We emphasize that although we assume independency of the single electrode processes in every state j, this does not mean that the electrodes are treated as independent. Correlations between electrodes are taken into account by the transitions between subsequent states. An HMM with output probabilities (6) is still a discrete HMM with a finite (binomial distribution) or infinite (Poisson distribution) alphabet since the symbol vector $\vec{k} = (k_1, \ldots, k_{30})$ consists of integer values $k_l = 0, 1, 2, \ldots$. The parametrization of the output function and the corresponding optimization procedure are rather in the spirit of continuous density HMMs (CD-HMMs) (Rabiner 1989; Huang et al. 1990) where one adapts, e.g., gaussian output densities. The Baum-Welch re-estimation formulas for Poisson and binomial output distributions are listed in the Appendix. We call HMMs based on this second approach PD-HMMs, where PD stands for parametrized (output) distribution.

3 Results

3.1 Preparation of the data

3.1.1 Relevant sections of the responses. The onset of the neuronal responses is not sharply connected with the moment at which the bar starts moving. This onset depends on the direction of the bar due to different distances of the starting point from, the receptive fields of the neurons. Therefore, we decided to define for each stimulus the relevant response interval symmetrically around the maximum of the time-dependent spike rate. This was obtained by summing the contributions from all 21 trials and 30 electrodes and a subsequent smoothing of the resulting spike signal with a 10-ms time window. In this way, we could determine a unique maximum for each of the 16 stimuli. We analyzed responses in time windows with a total duration of 30, 300, 500, and 800 ms.

3.1.2 Data segmentation. It is an open problem on which time scale relevant neuronal information is processed. Therefore, we investigated the data on different (coarse-grained) time scales. We divided the response intervals into a varying number of nonoverlapping time segments and counted the spikes in each bin. We made a systematic comparison of the two fundamentally different types of HMMs (VQ vs parametrized output probabilities), each with the two basic topologies (left-right vs ergodic models), and these in addition with a varying number of internal states. In this extensive investigation, we used time windows (response intervals) of lengths between 300 and 800 ms, which were divided into segments of 25, 50, and 100 ms length (see Table 3). This coarse-graining is suggested by the fact that there are only very few events even during a response to a stimulus. Neurons fire at a very low rate, which means, e.g., that a 50-ms segment contains at most 11 spikes, and that more than 50% of all

Table 1. Cross-classification for two runs of the parallel clustering algorithm ($|S| = 10$) with different initial group vectors (data: length 300 ms, bin width 50 ms). Shown are the number of spike-count vectors which are put into a group i during a first clustering and into a group j in a second clustering ($i, j = 1, \ldots, 10$). Thus, each row represents a group for the first run and each column, a group for the second run. If there were data-intrinsic clusters, one would obtain equivalent partitions, i.e., one would get only one single value in each row and column, so that each group of the first clustering could be identified with one group of the second clustering

	Groups of the second partition									
Groups of the	1	9	0	2	0	2	0	233	0	1
first partition	1	226	0	143	0	1	0	1	2	0
	43	5	65	6	9	24	0	9	47	13
	0	8	0	5	1	2	0	0	246	1
	68	1	11	5	36	5	1	10	45	6
	243	4	0	0	10	9	12	2	0	0
	0	0	0	1	1	0	119	0	0	0
	2	34	859	0	3	140	0	0	32	0
	6	0	0	9	236	0	1	0	0	0
	0	13	0	5	0	8	1	0	2	314

bins are empty or contain only one spike. We were, however, also interested in the performance of the HMMs on very short time scales. In this special investigation, we segmented a 30-ms interval around the time of maximal response into bins of only 2 ms.

As a result of the segmentation of the response intervals, a stimulus was represented as a sequence of 30-dimensional spike-count vectors, where each component was an integer value. We thus obtained sequences which varied in length between 5 for 500-ms time windows segmented into 100-ms bins and 32 for 25-ms bins and response intervals of 800-ms duration.

3.1.3 Vector quantization. As argued in Sect. 2.2.3, for VQ-HMMs one has to solve the problem of finding an alphabet of reasonable size $|S|$. This is achieved by vector quantization (VQ), where one divides all spike-count vectors into $|S|$ groups and identifies all vectors in one group with the same symbol. For up to $16 \times 21 \times 32 (= 10\,752)$ spike-count vectors which should be partitioned into groups, a maximal alphabet size in the range of 10–30 symbols was reasonable on statistical grounds. To accomplish this, we tested three methods: the first two below are standard methods of cluster analysis (unsupervised classification) (Duda and Hart 1973); whereas the third method is supposed to be especially suited for neuronal data.

Sequential cluster analysis. First, we tried the method of iterative optimization (or stochastic approximation), which can be interpreted as a winner-take-all neural network without neighborhood relations. Each group is represented by a 30-dimensional real-valued vector. Each spike-count vector belongs to the group for which the product of group vector and spike-count vector is maximal. To find optimal group vectors, one adds to the group vector each time a spike-count vector is attached to its group some multiple of the Euclidean distance between the group vector and the spike-count vector. This is repeated very often (each vector may be classified a thousand times), while turning the multiplication factor slowly down to zero. It was proven that this method divides the data into reasonable clusters (Tsypkin and Kel'mans 1967; Fu 1968). We found, however, that almost all spike-count vectors were put into the same group after several iterations independently of the initial values. In order to understand this, we projected the spike-count vectors into two dimensions. For this, we used principle component analysis and an Euclidean, distance-preserving, nonlinear method (Sammon 1969). There was no indication that these 2-dimensional spike-count vectors cluster into groups. Therefore, we can assume that the spike-count vectors are more or less uniformly distributed in 30-dimensional space. This appears to be the reason why stochastic approximation fails in partitioning the spike-count vectors.

Parallel cluster analysis. With this method, one tries to find a minimal distance partition. Again each group is represented by a 30-dimensional vector. One starts with some initial vectors. All spike-count vectors are attached to the group vector which has the shortest Euclidean distance to the spike-count vector. The average of all vectors of each group defines the new group vector. This is iterated until there are no more changes.

It is indeed possible to obtain groups of more or less equal size. As expected due to the largely unstructured distribution of the spike-count vectors, the partitions that arise depend on the initial values (Table 1). This implies that the allocation of the symbols is still arbitrary. This is of course not a disadvantage for our purpose, though one should keep in mind that the symbols may have no data-intrinsic meaning.

We repeated this method for $|S|$ varying from 8 to 22 in steps of 2. For all numbers of groups, the sum of the variances of the groups is more or less the same (Table 2). Therefore, we have no indication that one should use a special number of symbols, which is in accordance with the uniform distribution of the spike-count vectors.

Next we tested whether or not the chain of symbols still describes the stimulus. Taking the responses of length 500 ms segmented into 25-ms bins, and using standard classification methods, we get a rate of about 90% correct recognition (see Sect. 3.2.2). In contrast, using the chain of symbols (e.g., for $|S| = 20$) and replacing each symbol by its group vector, we can also apply

Table 2. Values for the quality of the partition. Shown is the sum of the variances of the groups for two different spike-count vector sets (1 = length 300 ms, bin width 30 ms; 2 = length 800 ms, bin width 50 ms) and for a varying size $|S|$ of the alphabet. There is no preferable number of groups, and therefore no natural alphabet

Number of groups $\|S\|$	Quality 1	Quality 2
8	23517	53949
10	21800	51043
12	21053	48968
14	20498	47510
16	19768	46110
18	19606	45357
20	19010	44469
22	19424	44688

standard classification methods to these data. As a result, we get 65% correct recognition, which is far less than the result with the original data. So we lose information about the stimuli in the clustering process.

Classification with thresholded count vectors. Finally, we have assumed that the actual number of spikes in a small time segment is not important and that it only matters whether this number is smaller or larger than some threshold value. This idea leads to the following algorithm for finding a finite alphabet. We set components of our spike-count vectors that are larger than 1 to 1. In this way, we obtain a chain of binary vectors describing a response to a stimulus. To take into account that there might be spikes due to noise, we also tested a higher threshold. We set all components with 1 or 0 spikes to 0 and all other components to 1. With these methods, one gets about 4000 and 2000 different vectors from the about 10 000 spike-count vectors in the beginning.

These are still too many. Therefore, we identify all binary vectors within a certain Hamming distance with the same symbol. The Hamming distance of two binary vectors is just the number of different bits. We do not use a sophisticated method for this procedure: We take a first vector and collect all vectors that have a Hamming distance less than our threshold, take a second vector and so on, until there is no vector left. With 4 or 5 bits as threshold for the Hamming distance, we again get an alphabet of about 20 symbols.

Unfortunately, we noticed that we lose even more information with this procedure than with the method of minimal distance partition. Applying standard methods of classification to the binary vectors, we obtain about 60% correct recognition, but only about 35% when we replace the binary vectors by the group vectors.

As a result of the above comparison, we decided to use the symbols obtained by parallel cluster analysis and arbitrarily chose $|S| = 20$ for a detailed investigation of the corresponding HMMs. We made sure by spot checking that all other alphabets led to only slightly different results. Although the above results indicate the absence of a natural alphabet, its existence is not fully excluded since, e.g., a variable and adaptive segmentation of the time series might lead to different conclusions.

3.2 Comparison of the models

3.2.1 Parametric vs VQ-HMMs. With the segmented and vector-quantized data, we are now in the position to obtain and compare various HMMs. In all cases, we used 11 repetitions of each stimulus for training the corresponding HMM and tried to classify the remaining 10 repetitions. The results of a systematic comparison with respect to the recognition rates of the two fundamental types of HMMs (PD-HMMs which according to Sect. 2.2.4 describe the spike-count vectors directly by a parametrized output probability density, and VQ-HMMs which use the vector quantized data) are listed in Table 3. The size of the error in this table (and also in Fig. 6, see below) is estimated to be about 5%, which for the high rates is distributed asymmetrically around the corresponding mean values. In this investigation, we assumed for the output probabilities of the PD-HMMs multivariate Poisson distributions according to (6, 7). We tested each type in the two basic topologies (see Fig. 2), left-right vs ergodic models, with a varying number of internal states and for various lengths of the time windows (response intervals) and segments. As a first result, we see that HMMs with a poissonian output probability always yield higher recognition rates than the corresponding VQ-HMMs. Second, left-right models are almost always superior to ergodic models. This is in accordance with the observation that the training of ergodic models often leads to models that are essentially linear and therefore close to left-right models. For example, the recognition rates obtained for left-right PD-HMMs trained on 300-ms response intervals lie up to 10% above the rates for corresponding ergodic PD-HMMs and 40%–50% above the rates for comparable VQ-HMMs. The last result apparently means that the information loss in vector quantizing the data is much more serious than the assumptions made for the output probabilities in PD-HMMs. For a better understanding, we investigated the segregation quality of the models obtained. Here we checked whether the 16 HMMs, one for each of the stimuli, are able to distinguish self-generated data. This means that we simulated or generated 20 new data sets with each HMM and classified these data in the same manner as the original data. The resulting self-classification rate is always 100% for our poissonian PD-HMMs. For VQ-HMMs, these rates vary between approximately 60% for long segments and almost 100% for short time segments. This implies that PD-HMMs are in general better adapted to the data than VQ-HMMs, although they contain a comparable number of free parameters. We also realized that the recognition rates decrease as the length of the response intervals is increased. This effect has a simple explanation. A closer look at the data shows that the time window, where the neuronal response is not compatible with spontaneous activity, has a length of about 300 ms. Therefore, longer response intervals contain more noise and are more difficult to classify. The dependence of the recognition rate on the segment length shows no systematic variation for bin widths between 25 and 100 ms. For longer segments, we find that the recognition rates

Table 3. Classification rates for PD-HMMs (a) and VQ-HMMs with $|S| = 20$ (b). The self-recognition rate for the VQ-HMMs (c) is explained in the text. The recognition rates marked with an asterisk were obtained for degenerate left-right models

a) Length (ms) / Bin width (ms)	Recognition rates for Poissonian PD-HMMs (%)								
	N	300		N	500		N	800	
		Left-right	Ergodic		Left-right	Ergodic		Left-right	Ergodic
25	12	90*		20	87*		32	83*	
	8	86	76	12	77	68	20	60	61
	4	86	78	6	69	73	10	65	60
50	6	85*		10	83*		16	83*	
	4	90	84	6	74	75	10	73	69
	2	76	77	3	72	69	5	60	64
100	3	91*		5	87*		8	82*	
	2	83	80	3	76	74	4	67	64
b) Length (ms) / Bin width (ms)	Recognition rates for VQ-HMMs (%) ($\|S\| = 20$)								
	N	300		N	500		N	800	
		Left-right	Ergodic		Left-right	Ergodic		Left-right	Ergodic
25	8	49	32	12	42	31	20	31	25
	4	40	34	6	25	31	10	30	29
50	4	48	36	6	37	30	10	28	28
				3	30	31	5	27	26
100				3	44	28	4	28	23
c) Length (ms) / Bin width (ms)	Self-recognition rates for VQ-HMMs (%) ($\|S\| = 20$)								
	N	300		N	500		N	800	
		Left-right	Ergodic		Left-right	Ergodic		Left-right	Ergodic
25	8	97	79	12	98	92	20	99	96
	4	84	80	6	88	85	10	98	89
50	4	86	67	6	84	81	10	98	91
				3	70	69	5	84	79
100				3	71	62	4	75	69

decrease in accordance with earlier results (Krüger and Becker 1991).

To obtain an impression of the quality of the typically obtained HMMs, we show in Fig. 4 examples of original and simulated responses to one of the stimuli. We used a PD-HMM with 12 states for the generation of the data on the right of this figure. This automaton is depicted in Fig. 5a. The spike patterns in each time bin were generated by randomly distributing spikes in each segment according to the Poisson law, (7), with the mean values prescribed by the HMM. For each repetition, a path through the HMM is chosen randomly, with probabilities determined by the transition matrix of the HMM. The recognition rates for the corresponding HMM ensemble are 77%, which is not the highest value reached. Nevertheless, we see that the overall patterns are quite similar. Note, however, that in the simulated data the transitions between low and high activity regions are not as sharp as in the original data (see, e.g., electrodes C2 and E2). This is due to the self-transitions on the states enforced by the chosen ratio of the number of segments ($T = 20$) over the number of states ($N = 12$). These transitions apparently produce a too large variability in the timing of the spike events, which is not present to this extent in the data. This is in accordance with the observation that the ratio T/N lies between 1.0 and 1.5 for models with the highest recognition rates of about 90%. The optimal models with $T/N = 1.0$ are treated separately in Sect. 3.2.2. The second automaton (depicted in Fig. 5b) is an example of the fact that one can considerably reduce the number of parameters in the HMMs if one is only interested in the recognition rate: This 3-state model represents the same stimulus but was trained on 100-ms segments. It contains essentially all the information necessary for the pattern recognition process, since with these models one obtains basically the same recognition rate (74%) as for the 12-state models. The same holds for all the other topologies and models we tested. We can thus get a quite compact representation of the ensemble of possible responses to a given visual stimulus. From a biological

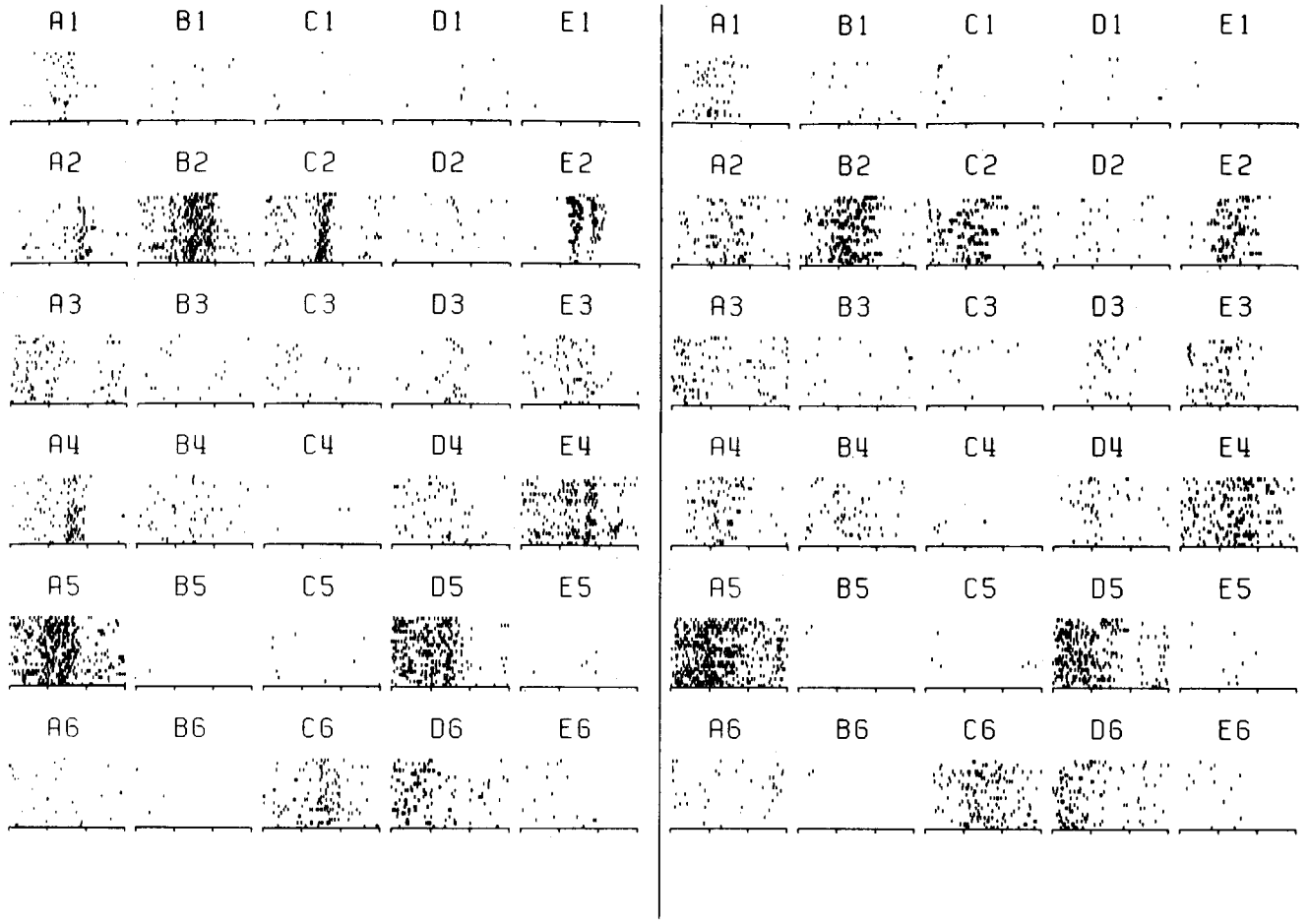

Fig. 4. Two dot displays of responses during 500 ms are shown. The *left* one is the original ensemble of responses to the same moving bar. The *right* one is a simulated response. The simulation was made with the strict left-right PD-HMM of Fig. 5a, based on 25-ms segments and a response interval of 500 ms. Different repetitions are shown below each other

point of view, it is interesting to note that the relevant time scale for the information processing appears to lie in the region around 25–100 ms.

3.2.2 Relation to linear classifiers. At this stage it is appropriate to remember that the recognition rates with simple linear classifiers lie around 90% (Krüger and Becker 1991)[1], which is similar to the best results reported above. It is explained below that linear classifiers can be understood as limiting cases of HMMs, namely models with transitions to only one next state (i.e., left-right models as in Fig. 5a, with the delay loops missing). Therefore, it should be possible to obtain similar recognition rates with such degenerate HMMs. This is indeed the case: If the number of nodes in our left-right models coincides with the number of segments in the chosen time window, we get the highest recognition rates. For example, for a 3-state model based on 300-ms data windows segmented into 100-ms bins, we obtain 91% correct classifications. The dependence of the recognition rate on the segment length is depicted for these models in Fig. 6.

This result deserves several comments. Let us first explain the equivalence of these degenerate HMMs with linear classifiers: The simple linear structure of the HMMs ($a_{ij} = \delta_{j,i+1}$, for $i = 1, 2, \ldots, T-1$, with $T =$ number of time segments = number of HMM nodes) and the assumed factorization of the output probabilities (6) imply that the probability for the spike-count vector sequence $\vec{k}(1), \vec{k}(2), \ldots, \vec{k}(T)$ factorizes with respect to discrete time t and with respect to the electrode number, i.e.,

$$P(\vec{k}(1), \vec{k}(2), \ldots, \vec{k}(T)) = \prod_{t=1}^{T} \prod_{l=1}^{30} P(k_l(t), \mu_i^{(l)}) \qquad (9)$$

Such an HMM can be regarded as the representation of a multivariate probability distribution of $(30 \times T)$-dimensional spike-count vectors $\vec{k}$ around the prototype vector of mean rates $\vec{\mu}$, with components $\mu_i^{(l)}$. Since the factorization implies that the covariance matrix is diagonal, our data are now fully characterized by the 16 prototype vectors $\vec{\mu}_i$, $i = 1, \ldots, 16$ and the corresponding Poisson

[1] The recognition rates reported in this paper increase by about 10% for the data set used in the current work. See also Fig. 6.

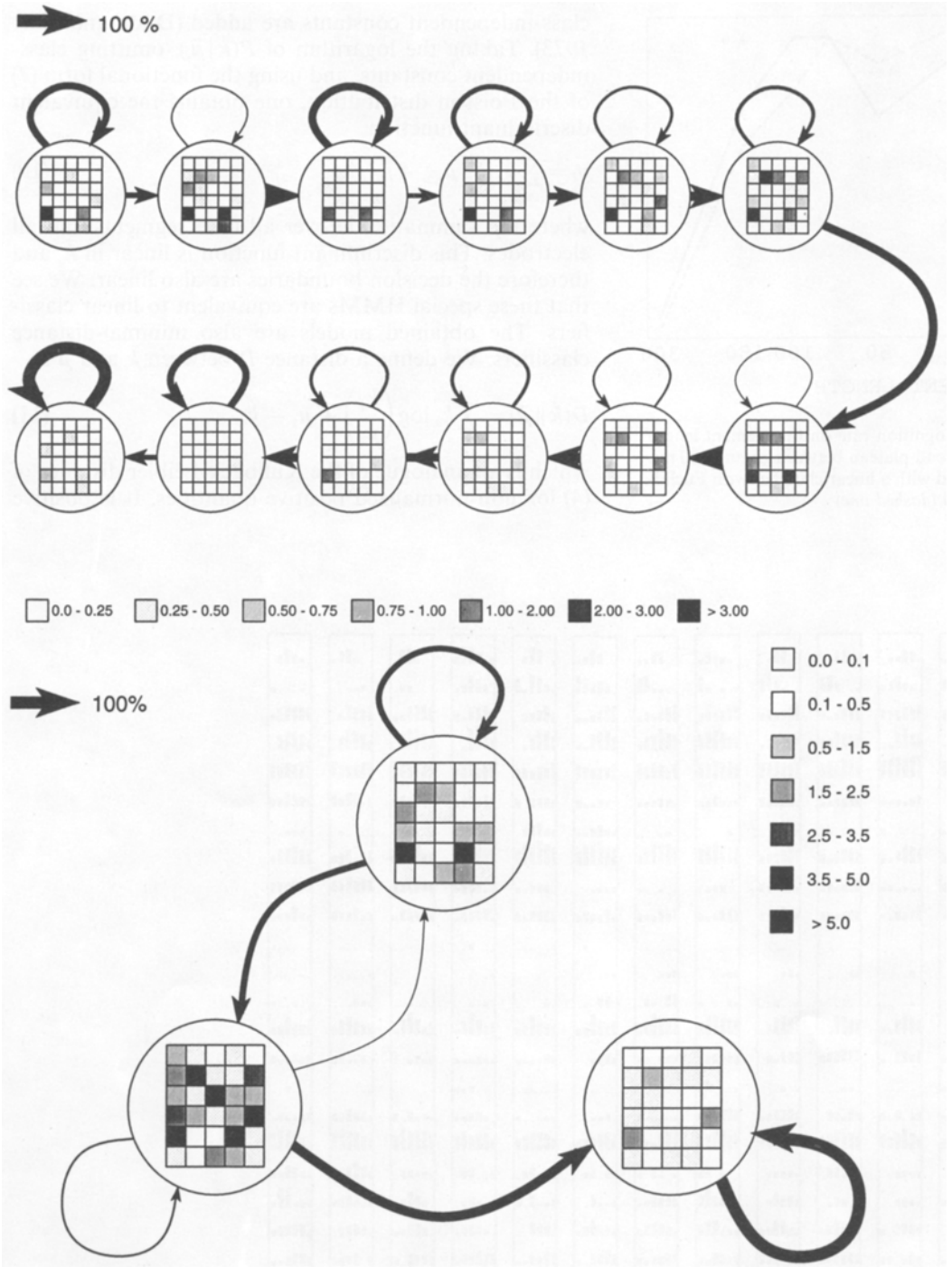

Fig. 5. Two typical examples of obtained PD-HMMs. **a** (*top*) this 12-state left-right model was used to generate the data on the right of Fig. 4. **b** (*bottom*) an ergodic 3-state model which describes the same stimulus response as in **a**, but on a coarse-grained time scale. It starts with $\pi_1 = 91\%$ in the state drawn on the *top*. The gray levels of the inserts code for the mean rates at the electrodes (arranged as in the experimental setup), if the system is in the corresponding state. The width of the *arrows* measures the transition probabilities between the states. The HMM types yield comparable recognition rates on 500-ms data samples

distributions. As one would expect, the mean rates estimated with the Baum-Welch formula (see the Appendix) coincide in this case with the rates in each time segment obtained from a simple average over the training samples. The resulting prototypes are depicted in Fig. 7. As explained in Sect. 2.2.2, the probabilities of (9), which are conveniently denoted by $P(\vec{k} \mid \vec{\mu}_i)$, serve as discriminant functions $g_i(\vec{k})$ in the classification process with HMMs, i.e., an observed total spike-count vector $\vec{k}$ is classified into class number i for which $g_i(\vec{k})$ is largest. The classification process is not altered if a monotonically increasing function is applied to all g_i, or if

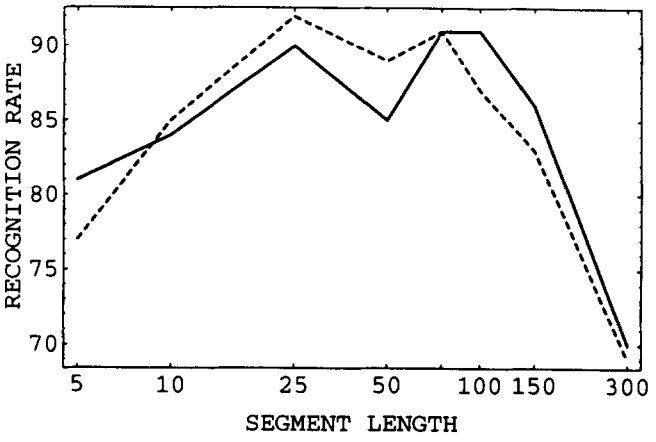

Fig. 6. The dependence of the recognition rate on the segment length for degenerate HMMs shows a broad plateau between 25 and 100 ms. For comparison, the rates obtained with a linear classifier with Euclidean distance measure are depicted (*dashed line*)

class-independent constants are added (Duda and Hart 1973). Taking the logarithm of $P(\vec{k} \mid \vec{\mu}_i)$, omitting class-independent constants, and using the functional form (7) of the Poisson distribution, one obtains the equivalent discriminant function

$$g_i = \sum_\alpha k_\alpha \log \mu_{i,\alpha} - \mu_{i,\alpha} \tag{10}$$

where the summation is over all time segments and all electrodes. This discriminant function is linear in $\vec{k}$, and therefore the decision boundaries are also linear. We see that these special HMMs are equivalent to linear classifiers. The obtained models are also minimal-distance classifiers: We define a distance D between $\vec{k}$ and $\vec{\mu}$ as

$$D(\vec{k} \| \vec{\mu}) = \sum_\alpha k_\alpha \log\left(\frac{k_\alpha}{\mu_\alpha}\right) + \mu_\alpha - k_\alpha \tag{11}$$

which is an analogue of the Kullback-Leibler distance of (4) for non-normalized positive quantities. It is positive

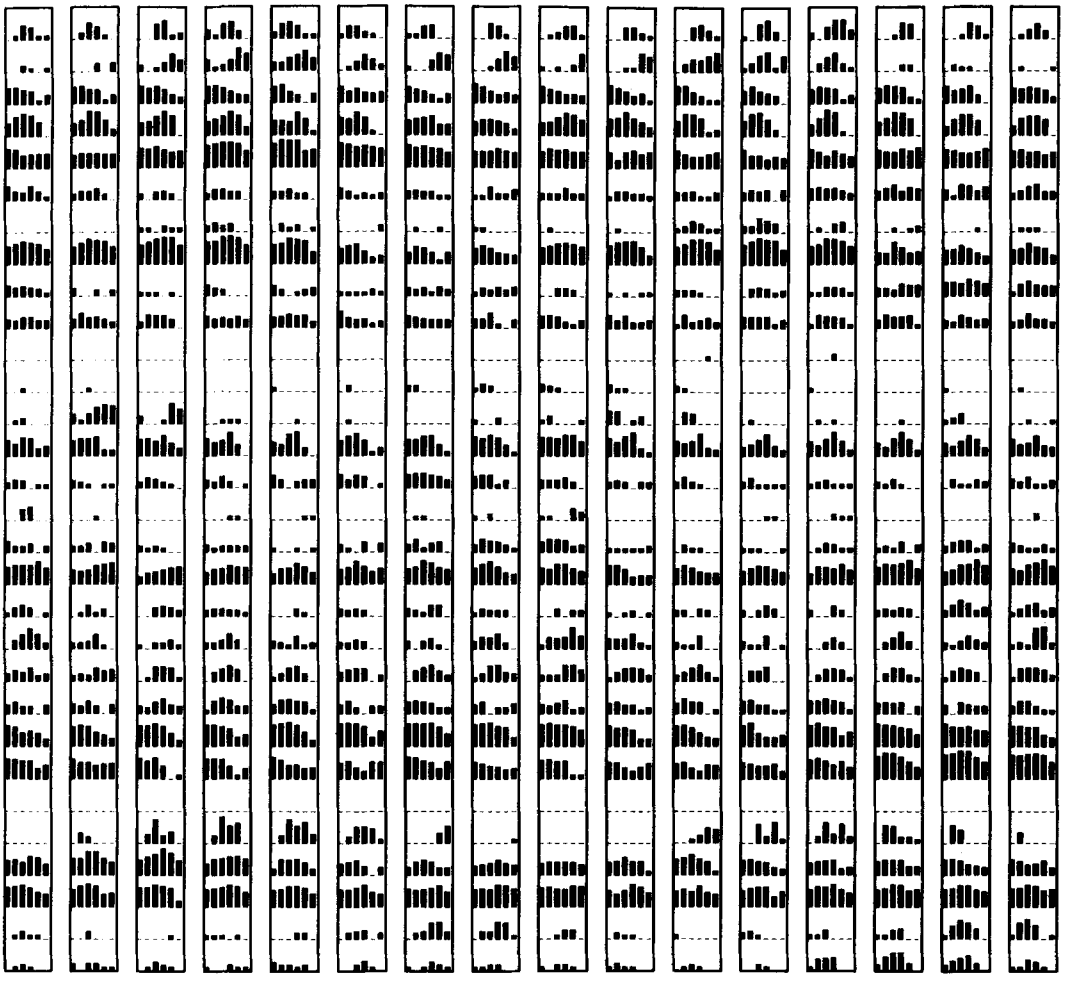

Fig. 7. The neuronal responses to the 16 stimuli of the experiment are optimally classified by the 16 prototype vectors shown as columns. *From left to right*, the angle of the bar increases in equal steps. The components of each prototype vector are the time-dependent mean rates in each segment at every electrode (here 6 time segments per electrode, electrodes from top to bottom: *A1, A2,. . ., E6*). The height of the bars in each segment measures the corresponding rates on a logarithmic scale. The logarithmic scale is motivated by (10, 11)

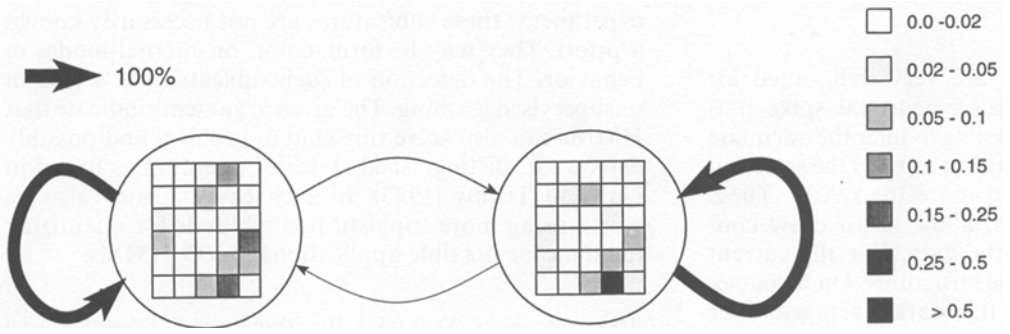

Fig. 8. A typical HMM obtained for short time windows (30 ms) and 2-ms segments. This is basically the superposition of two binomial processes since the transitions between the states can be neglected. For about half of the stimuli, the two states can be reduced to one state

and zero only if $\vec{k}$ and $\vec{\mu}$ coincide. It is easily seen that the same classification of a pattern $\vec{k}$ is achieved by searching for the prototype μ_i for which $D(\vec{k} \| \vec{\mu}_i)$ is minimal. In contrast, the linear classifiers used in Krüger and Becker (1991) utilized the Euclidean distance measure $|\vec{k} - \mu_i|$, which is known to correspond to multivariate gaussian distributions with identical diagonal covariance matrices for all classes (Duda and Hart 1973). Since we are dealing with point processes and therefore with discrete events a continuous probability density as the output process of an HMM is not appropriate from a physical point of view, although it leads (with an optimally chosen variance parameter) to classification rates similar to those of the more adequate Poisson model (see Fig. 6).

The fact that these degenerate HMMs yield the best results for almost all time windows and segmentations implies that the data have a quite simple structure on time scales between 25 and 100 ms. Firstly, from (9) it follows that to a very good approximation the fluctuations on different time segments and on different electrodes are independent. This implies that the corresponding covariance functions are zero on these time scales. A direct numerical computation confirmed this result. Secondly, the data for each visual stimulus are well described by one assigned prototype spatiotemporal excitation pattern. Such a pattern consists, e.g., of a time sequence of six rates (data averaged over 50 ms) for each electrode, as depicted in Fig. 7. As mentioned above, an averaging over more than 100 ms results in a decrease of the recognition rates. Thirdly, the pattern space turned out to be linearly separable, which explains the success of linear classifiers as used in Krüger and Becker (1991). As will be seen below, this picture is no longer valid if we go to small time scales. It is also obvious that with these models one can reproduce the original data, e.g., of Fig. 4, with high accuracy if the segmentation is chosen small enough. In particular, the transitions between high and low activity regions become sharper due to the missing delay loops.

3.2.3 Performance on short data segments. From a biological point of view, it is interesting to know whether a stimulus can be recognized by using only a relatively small part of the elicited spike patterns. This addresses the question of whether it is possible for an animal to make reliable decisions in short times. Therefore, we decided to use a window of only 30-ms length around the maximal response, which was binned into segments of 2-ms length. We observed at most one spike event in each segment. For such processes, it is appropriate to use the binomial distribution of (8) with $n = 1$, i.e., a Bernoulli process, as output on the states of the HMMs. By varying the number of nodes of the HMMs, it turned out that two states yield optimal recognition rates. Further, we found that in about half of the 16 stimuli the HMMs reduce or are equivalent to one state only, which means that in these cases the response is well characterized by a single mean rate for each electrode. In the remaining cases, the transitions between the two states are often very low, and therefore the corresponding process is basically a superposition of two Bernoulli processes. An example of such an HMM is depicted in Fig. 8. Here the initial probabilities are 0.5 for each state, which essentially means that one-half of the training samples is characterized by the 30 mean rates of node 1, and the other half by the rates of node 2. Thus, there is not much temporal structure on the time scale of 30 ms, and the corresponding HMMs are again quite simple, although not of the linear type as above. The two states suggest that during the experiment the animal was in two different response modes. One simple physiological explanation of this effect is a change of the monkey's eye position in the course of the stimulus repetitions. Such spontaneous eye movements may occur even with paralyzed animals, with the result that different receptive fields and correspondingly different neuronal excitation patterns are elicited by the same visual stimulus. For two stimulus classes, we also found non-negligible transition probabilities between the two states, which means that the rates change within the 30-ms time window with a certain probability. From such a model, one can also extract the mean dwell time in each state, which amounts to an automatic segmentation of the data. The latter models show that there is variability in the timing of the events on this time scale.

The classification rates with these models lie at approximately 50%, which is 10%–20% more than the rates for corresponding linear classifiers. This shows that for only slightly more complex excitation patterns, the HMM algorithms are superior to conventional pattern recognition algorithms. This is important since we expect that data from future experiments with active monkeys will have a higher degree of complexity than those investigated in this paper.

4 Discussion

We have shown that HMMs are very well suited for classifying, coding, and analyzing neuronal spike patterns. It turned out that it is possible to infer the adequate class of models by using recognition rates as the selection criterion. We obtained rates of more than 90%. These high rates, on the other hand, allow us to draw conclusions about the nature of the data. For the current data, we found relatively simple structures: On a coarse-grained time scale, i.e., for the data averaged over segments of 25–100 ms length, we found that the neuronal responses to the stimuli are well described by spatio-temporal prototype patterns of mean rates with a poissonian statistic superimposed. Further, the pattern classes appear to be linearly separable, which explains the success of our degenerate HMMs and of the corresponding linear classifiers. The fact that the recordings at a given electrode may contain contributions from more than one cell does not affect the above conclusions. In general, however, the interpretation of the extracted HMMs may depend on the number of simultaneously observed data sources.

By taking the pattern samples in small time windows of only 30 ms, we still found recognition rates of about 50%, which is a factor of eight higher than an unbiased guess. This implies that, in principle, decision tasks can still be solved with reasonable accuracy on these time scales. Our investigations also revealed that there is a higher degree of variability in the better resolved data. At the moment, the small number of data samples forms the main limitation to obtaining a more accurate characterization of the corresponding probabilistic structures. An interesting point in this context is that the addition of more electrodes will not necessarily enhance the discrimination ability with respect to the stimulus classes. On the contrary, by selecting the 20 most important electrodes, we basically obtain the same recognition rates as with the full set of 30 electrodes, which on the other hand means that the HMMs are robust against noise or irrelevant information. The importance of individual electrodes with respect to the recognition task was determined in Dülfer (1993) with standard methods of variable selection from statistical classification theory (see e.g., Chap. 6 of Hand 1981).

Experiments with permanently implanted electrodes are currently being prepared which will provide us with much larger data sets. Based on the results reported in this work, we expect that HMMs will be very useful in evaluating such experimental data. It is expected that this will be even more the case if the responses in one class are more structured and differentiated due to, for instance, the presence of several features represented in the responses. With the current data, such a situation can be simulated by neglecting, e.g., the information about the direction of the moving bar, while retaining knowledge about the orientation. This results in eight classes corresponding to eight orientations in the experiment. Each extracted HMM in such a situation consists of two branches, which code the feature that each orientation allows for two signs of the velocity. In the planned experiments these subfeatures are not necessarily known a priori. They may be form, color, or internal modes of behavior. The detection of such subfeatures is a task in unsupervised learning. The given arguments indicate that HMMs can also solve this kind of problem and possibly detect, e.g., distinguished states of attention as found in Gat and Tishby (1993). In such cases, it may also be worth using more sophisticated methods for quantizing the data for possible applications of VQ-HMMs.

Acknowledgements. We thank J. Honerkamp and V. Breuer for useful discussions, and the latter also for help with some of the figures. This work was partially supported by the Deutsche Forschungsgemeinschaft (Schwerpunkt 'Physiologie und Theorie Neuronaler Netzwerke').

Appendix

The parameters $\vec{\lambda} = (\vec{\pi}, A, \{B(S)\})$ of the HMMs (see Sect. 2.2.1) are calculated iteratively. To avoid too many indices we mark the parameters from the next iteration by a hat – e.g., $\hat{a}$ – and leave the parameters from the preceding iteration step unmarked. As usual we define the following useful quantities (see e.g. [Rabiner 1989, Huang et al., 1990]):

$$\alpha_j(t) := \begin{cases} \pi_j b_j(O_1) & \text{for } t = 1 \\ \sum_{i=1}^{N} \alpha_i(t-1) a_{ij} b_j(O_t) & \text{otherwise} \end{cases},$$

$$\beta_i(t) := \begin{cases} 1 & \text{for } t = T \\ \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_j(t+1) & \text{otherwise} \end{cases},$$

$$\psi_{ij}(t) := \frac{a_{ij}\alpha_i(t)\beta_j(t+1)b_j(O_{t+1})}{P(O \mid \vec{\lambda})},$$

$$\Psi_i(t) := \sum_j \psi_{ij}(t) = \frac{\alpha_i(t)\beta_i(t)}{P(O \mid \vec{\lambda})}.$$

In our work the observation sequence $O = O_1 O_2 \ldots O_T$ is either a sequence of symbols $S_t \in \{S\}$, $t = 1, \ldots, T$ in the case of VQ-HMMs, or a sequence of spike-count vectors $\vec{k}(t)$ for PD-HMMs. The probability $P(O \mid \vec{\lambda})$ is defined in (1) and can be calculated as

$$P(O \mid \vec{\lambda}) = \sum_i \alpha_i(T)$$

The quantities α and β are known as forward respectively backward probabilities. For both types of models one obtains the following iteration equations for the elements of the transition matrix A and the initial probability $\vec{\pi}$:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \psi_{ij}(t)}{\sum_{t=1}^{T-1} \Psi_i(t)}$$

$$\hat{\pi}_i = \Psi_i(1)$$

For VQ-HMMs the EM-algorithm yields the following recursion for the output probabilities b:

$$\hat{b}_i(S) = \frac{\sum_{t=1}^{T} \Psi_i(t)\delta_{S,o_t}}{\sum_{t=1}^{T} \Psi_i(t)}$$

and for PD-HMMs with Poisson densities one obtains for the update of the mean values μ [Becker 1994; Dülfer 1993]

$$\hat{\mu}_i^{(l)} = \frac{\sum_{t=1}^{T} \Psi_i(t)k_l(t)}{\sum_{t=1}^{T} \Psi_i(t)}.$$

For binomial output densities $\hat{\mu}_i^{(l)}$ has to be replaced by $\hat{\mu}_i^{(l)} \cdot n$. In the case of multiple observation sequences one has to sum over all observations in each enumerator and denominator.

References

Bahl LR, Jelinek F, Mercer RL (1983) A maximum likelihood approach to speech recognition. IEEE Trans Pattern Anal Machine Intell 5:179–190

Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–171

Becker JD (1994) Versteckte Dynamik neuronaler Prozesse. Harry Deutsch Verlag, Frankfurt

Becker JD, Honerkamp J, Hirsch J, Schlatter E, Greger R (1994) Analyzing ion channels with hidden Markov models. Pflügers Arch 426:328–332

Chung SH, Moore JB, Xia L, Premkumar LS, Gage PW (1990) Characterization of single channel currents using digital signal processing techniques based on hidden Markov models. Philos Trans R Soc Lond [Biol] 329:265–285

Chung SH, Krishnamurthy V, Moore JB (1991) Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents buried in noise and deterministic interferences. Philos Trans R Soc Lond [Biol] 334:357–384

Dempster AP, Laird NM, Rubin DR (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc [B] 39:1–22

Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York

Dülfer B (1993) Klassifikation and Merkmalsextraktion. Thesis, University of Freiburg

Forney GD (1987) The Viterbi algorithm. Proc IEEE 61:268–278

Fredkin DR, Rice JA (1992) Maximum likelihood estimation and identification directly from single-channel recordings. Proc R Soc Lond [Biol] 249:125–132

Fu KS (1968) Sequential methods in pattern recognition and machine learning. Academic Press, New York

Gat I, Tishby N (1993) Statistical modelling of cell-assemblies activities in associative cortex of behaving monkeys. In: Moody JE, Hanson SJ, Lippmann RP (eds) Advances in neural information processing systems 5. Morgan Kaufmann, San Mateo, p 945

Gray CM, König P, Engel AK, Singer W (1989) Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. Nature 338:334–337

Gray RM (1984) Vector quantization. IEEE Acoust Speech Signal Process Mag 1:4–29

Hand DJ (1981) Discrimination and classification. Wiley, Chichester

Huang XD, Ariki Y, Jack MA (1990) Hidden Markov models for speech recognition. Edinburgh University Press, Edinburgh

Krüger J, Aiple F (1988) Multimicroelectrode investigation of monkey striate cortex: spike train correlations in the infragranular layers. J Neurophysiol 60:789–828

Krüger J, Becker JD (1991) Recognizing the visual stimulus from neuronal discharges. Trends Neurosci 14:282–286

Makhoul J, Roucos SR, Gish H (1985) Vector quantization in speech coding. Proc IEEE 73:1551–1558

Pawelzik K, Bauer HU, Deppisch J, Geisel T (1993) How oscillatory neural responses reflect bistability and switching of the hidden assembly dynamics. In: Moody JE, Hanson SJ, Lippmann RP (eds) Advances in neural information processing systems 5. Morgan Kaufmann, San Mateo, p 977

Paz A (1971) Introduction to probabilistic automata. Academic Press, New York

Rabiner LR (1989) A tutorial on hidden Markov models and selected application in speech recognition. Proc IEEE 77:257–285

Radons G, Becker JD, Dülfer B (1992) Analysis of multielectrode data with hidden Markov models (Abstract). In: Loose W (ed) 18th IUPAP International Conference on Statistical Physics. Berlin, p 175

Richmond BJ, Optican LM, Podell M, Spitzer H (1987) Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex. I. Response characteristics. J Neurophysiol 57:132–146

Sammon JW (1969) A nonlinear mapping for data structure analysis. IEEE Trans Comput 18:401–409

Tsypkin YaZ, Kel'mans GK (1967) Recursive algorithms of self-learning. Izv Akad Nauk SSSR Tekhn Kibernetika (Engineering Cybernetics SSSR) 5:70–80