

Hidden Markov Models in the Neurosciences

Blaettler Florian, Kollmorgen Sepp,
Herbst Joshua and Hahnloser Richard
*Institute of Neuroinformatics,
University of Zurich / ETH Zurich
Switzerland*

1. Introduction

HMMs are useful tools for model-based analyses of complex behavioral and neurophysiological data. They make inference of unobserved parameters possible whilst taking into account the probabilistic nature of behavior and brain activity. The trend in neuroscience is to observe and manipulate brain activity in freely moving animals during natural behaviors and to record from several dozens of neurons at the same time. The richness of the data generated in such experiments constitutes a major challenge in data analysis, a challenge that can be addressed partly using HMMs. For example, an experimenter could be interested in recovering from noisy measurements of brain activity the underlying electrical activity of single brain cells using the constraint that activity has to agree with cellular biophysics (such as in the spike sorting problem). Or, the experimenter may want to describe the variance in some behavior and relate it to causes encoded in the neural recordings. We here review recent HMM applications to illustrate how HMMs enhance the experimental read out and provide insights into neural population coding. We discuss HMMs for identifying repetitive motifs in natural behaviors, in particular birdsong and for extracting and comparing patterns of single-neuron activity in multi-neuron recordings. Finally, we introduce a pair HMM for sequence alignment that is free of distance measures and aligns sequences by matching their self-similarities. We demonstrate the workings of the new pair HMM by aligning the song of a pupil bird to that of its tutor.

2. General overview

HMMs are widely applied in neuroscience research, ranging from studies of behavior, to neuron assemblies, and to individual ion channels. The observed variable (the output of the HMM) can be a test subject's decision, or an animal's motor output. In other cases, the observed variable is neural activity measured using electrophysiology, electroencephalography (EEG), magnetoencephalography (MEG), or imaging. What many studies have in common is the quest to identify underlying brain states that correlate with the measured signals. Also, many studies create the need of segmenting behavioral or neural sequences into recurring elements and transitions between them.

2.1 Decoding of neural data with HMMs

Spike data recorded from single or multiple nerve cells (neurons) is amenable to modeling with HMMs. Neurons emit action potentials. These are brief and stereotyped electrical events that can be recorded with extracellular electrodes in behaving animals. A popular application of HMMs is decoding information from recorded spike data. For instance, Pawelzik et al. (Pawelzik, Bauer et al. 1992) used an HMM to model neuronal responses in the cat's visual cortex. Their model distinguishes periods of oscillatory versus stochastic firing. Gat and Tishby (Gat and Tishby 1993) modeled monkey cortical activity as a multivariate time-dependent Poisson process. The Poisson means are hidden parameters. The recorded spike trains were divided into small time bins in which the spike counts were assumed to obey a Poisson distribution with time-dependent mean rate. Gat and Tishby's model yielded a temporal segmentation of spike data into a sequence of 'cognitive' states, each with its distinguished vector of Poisson means. Along the same lines, Radons et al. (Radons, Becker et al. 1994) employed HMMs for decoding the identity of visual stimuli from recorded neural responses. They simultaneously recorded neuronal spiking activity from several neurons in the visual cortex of monkeys during presentation of different visual stimuli. For each stimulus, they trained an HMM on a subset of the respective trials. HMM outputs were formed from the neural activity of the simultaneously recorded cells, assuming Poisson spike statistics. After learning, the hidden states of the HMMs corresponded to various regimes of multi-unit activity. Radons et al. then used the trained HMMs to decode stimulus identity from neural responses by selecting the stimulus for which the HMM gives the largest likelihood for generating the neural responses. Using this procedure they were able to identify the presented visual stimulus with high accuracy.

Related to these studies, several authors have investigated the idea of cell assemblies and their associated sequential or attractor dynamics. A state of a cell assembly is given by a particular pattern of activity in that assembly. States and transitions of assemblies are thought to bear functional relevance and can be efficiently modeled using HMMs (Gat, Tishby et al. 1997; Nicolelis, Fanselow et al. 1997; Rainer and Miller 2000). Thereby, insights can be gained into the mechanisms of neuron firing under different pharmacological conditions (Camproux, Saunier et al. 1996). Also, rich patterns of neural activity have been observed during sleep; these patterns often resemble motor-related activity during the day and are thought to constitute a form of replay activity. Such replay activity has, for example, been observed in songbirds (Dave and Margoliash 2000; Hahnloser, Kozhevnikov et al. 2002; Weber and Hahnloser 2007). HMMs have helped to gain insights into such activity patterns by segmenting them into discrete global states. One successful method for doing this is to assume that for each global (hidden) state of the neuron population, neurons have renewal firing statistics determined by the probability density of their interspike intervals (Camproux, Saunier et al. 1996; Danoczy and Hahnloser 2006). By training an HMM on the simultaneously recorded spike trains, each model neuron learns to fire spike patterns with renewal statistics that can be different for each of the hidden states. Using this procedure, it was found that sleep-related activity in songbirds is characterized by frequent switching between two awake-like firing states, one in which the bird is singing, and one in which it is not singing (Danoczy and Hahnloser 2006; Weber and Hahnloser 2007).

Several authors have explored HMM based decoding of neural or nerve activity for control purposes, for example, to control a robot arm. Chan and Englehart (Chan and Englehart 2005) recorded myoelectric signals from the forearm during six different kind of movements, represented by six hidden states in their HMM. Their goal was to infer the

correct arm movement (the correct hidden state) from the recorded myoelectric signal. Knowing the regular dynamics of limb movements, and to avoid overfitting, the authors constrained the transition matrix of the HMM down to a single parameter α , the probability of remaining in the same hidden state in two consecutive time steps (32 ms). All transitions to other states were equally probable. Emission distributions were modeled as Gaussians; their parameters were directly estimated from the training data. Using this model, Chan and Englehart reach a classification rate of 94.6% correct, which exceeds the performance of other algorithms. One of the advantages of using a probabilistic model is that it allows, through continuous training, to continuously adapt to long term system changes, such as changes in the electrode-skin interface. Other neuroprosthetics work relies solely on neural recordings to control the robotic arm. Recordings are usually made in brain areas that are responsible for motor planning or motor commands. The activity of recorded cells is decoded and related to a certain motion of the robot arm. Abeles and others (Abeles, Bergman et al. 1995; Seidemann, Meilijson et al. 1996; Gat, Tishby et al. 1997) analyzed neural population data from the frontal cortex of monkeys performing a delayed localization task. The monkeys were trained to perform a delayed instructed arm movement. This task made it possible to record from neurons during the planning phase in which the monkey knew what to do, but was not moving yet, and during the movement phase itself. In such experiments it is possible to use an HMM to estimate the upcoming motor output from the recorded neural signal during the planning phase and control the prosthetic robot arm using decoded planning-related activity. Poisson firing statistics are usually assumed whereby the mean firing rate depends on the current state of the neuron population that is hidden from the observer. Movements or movement intentions can be decoded in such scenarios for example by thresholding the posterior probabilities of hidden states (Kemere, Santhanam et al. 2008).

HMMs have also been used to model neural activity on a much smaller scale. Action potentials are formed by different ion channels that can either be closed or open, i.e. permeable for a special type of ion or not. The state of single ion channels can be recorded using cell-attached recording modes. The state (open or closed) of a single ion channel is probabilistic and depends on its own history and the history of membrane voltage amongst other factors. HMMs can model the dynamics of single ion channels and be used to estimate their state trajectory from noisy recordings (Chung, Moore et al. 1990; Becker, Honerkamp et al. 1994).

2.2 HMMs as tools in data analysis

In contrast to the abovementioned cases in which HMMs are used to directly model a hidden parameter of interest, HMMs are also commonly used as intermediate steps in data analysis, e.g. artifact correction. Dombeck et al. (Dombeck, Khabbaz et al. 2007) describe an experimental apparatus for two-photon fluorescence imaging in behaving mice; the mice are head-restrained while their limbs rest on a Styrofoam ball. The mice maneuver on the spherical treadmill while their head remains motionless. In such experiments it is common to observe running-associated motion artifacts in the focal plane of the microscope. The displacement of the brain relative to the microscope throughout the scanning process can be described by a (hidden) random walk on a finite two-dimensional grid. Key to artifact correction is the fit of the scanned image at a given time point and displacement compared to the reference image. The parameters of the displacement can be learned by maximizing over the joint probability of displacements and image similarities with the expectation

maximization algorithm. After correcting the motion artifacts in the image sequence, the cleaned data can be further analyzed to study the neural code or other questions of interest.

2.3 Analysis of psychophysical data with HMMs

HMMs have been used to infer when learning occurs in behaving subjects and shown to provide better estimation of learning curves than other methods. Smith et al. (Smith, Frank et al. 2004) studied learning in a binary choice task – subjects had to make a choice out of two possibilities (correct versus incorrect). In this case the observed data are a time series of Boolean values. The authors assumed that the subject's answers followed a Bernoulli distribution that depends on a hidden state, reflecting the subject's performance. Using this latent variable model they quantified the probability that subjects performed better than chance, as follows. The first estimated the hidden learning dynamics, which allowed them to estimate the subject's performance on a fine timescale, essentially on a trial by trial basis. Using a confidence bound on the inferred learning curve they estimated the exact trial when learning has occurred. This trial happened usually much earlier than when determined using less elaborate methods, revealing the superiority of the hidden-state approach.

2.4 Analysis of natural behavior with HMMs

Natural behaviors are increasingly the target of neuroscience research but they are much more difficult to characterize than controlled behaviors because of their many inherent degrees of freedom. In typical experiments, natural behaviors are captured by means of movies or sound recordings, or by placing sensors or emitters on critical body parts. It is often difficult to classify natural behaviors. For example, to classify the swimming behaviors of fish, or the mating behaviors of flies, human experimenters usually must painstakingly analyze the various image frames and inspect them for repetitive movement patterns. Obviously, it would be much more convenient to automate such processes and let machines do the pattern extraction. Today, efforts are underway to develop such techniques and HMMs are a key methodology with great potential.

In the following, we introduce HMMs for analyses of complex vocal output, namely the songs of songbirds. The birdsong analysis problem bears resemblance with the speech recognition problem. However, the nature of birdsong learning creates different challenges for birdsong analysis. Therefore, different HMM approaches may be required as we will see next.

3. Alignment of birdsong with HMMs

Songbirds learn their songs from a tutor early during life, much like children learn their mother tongue from their parents. Songs in many species are composed of repetitions of a song motif that is composed of several syllables. The song motifs and syllables in closed-end learners such as the zebra finch are very stereotyped and do not change much during adulthood (Immelmann 1969). Song development, on the other hand, is more complex, and at any time juvenile birds can alter the spectral or temporal features of their songs: young birds can morph a syllable into another, drop an existing syllable, or introduce new syllables (Tchernichovski, Nottebohm et al. 2000; Tchernichovski, Mitra et al. 2001; Gardner, Naef et al. 2005). During development, song syllables may change independently of each other. Furthermore, songs can vary in speed from rendition to rendition (Gardner, Naef et al. 2005; Glaze and Troyer 2006).

A standard problem in birdsong analysis is that of segmenting the songs into motifs and into syllables. HMMs are useful for automated segmentation of birdsongs (Kogan and Margoliash 1998) because they can adequately deal with variable song tempo and spectral variability of song syllables.

However, in many circumstances song analysis goes beyond segmentation. A typical situation for a birdsong researcher is that he or she wants to compare the song of a juvenile bird to that of his tutor, to find out how much the pupil has already learned. To track song development through time, corresponding song elements have to be reliably identified across developmental phases. Below, we illustrate the use of HMMs for song comparison. The goal is twofold: first, it is to measure the similarity between two songs in terms of a single scalar quantity; and, second, to identify matching elements in the two songs. Assessment of song similarity has proven to be a very useful tool for developmental studies in which enabling or disabling influences on song learning are studied (Tchernichovski, Nottebohm et al. 2000; Tchernichovski, Mitra et al. 2001; Gardner, Naef et al. 2005; London and Clayton 2008). The identification of matching song elements between different birds is of relevance in multi tutor studies in which we would like to trace back song elements to the tutor they have been learned from (Rodriguez-Noriega, Gonzalez-Diaz et al. 2010).

To compare two different songs with each other bears resemblance with comparing the genomes of two different species, in which insertions or deletions of long strands of DNA sequences are frequently encountered. The problem of finding correspondences between sequences is often referred to as the alignment problem (Brown, Cocke et al. 1990; Durbin 1998). Clearly birdsong alignment is different from genome alignment, because in the former both spectral and temporal features change during development, whereas there is no analogy of spectral changes in genomes (the four letter alphabet of nucleotides has been preserved by evolution).

3.1 Pair HMMs for birdsong alignment

Computational approaches to the alignment problem, like minimal edit distance algorithms (Wagner and Fischer 1974), have been around for quite some time and recently pair hidden Markov models (pair HHMs) have become very common. They offer a unified probabilistic framework that entails these more basic techniques but is much more general (Durbin 1998). One advantage of pair HMMs in alignment over more standard dynamic programming techniques is that pair HMMs do not require ad-hoc parameter setting: the trade-off between insertions, deletions, and matches can be learned from the data and does not have to be set by hand. Before we introduce a new pair HMM architecture and apply it to birdsong alignment, we first illustrate the general problem of alignment using a toy example.

Consider the following two sequences: *ABAC* and *AADC*. The second sequence results from the first by deleting *B* and inserting *D* at a different location. A possible way to describe the relationship between the two sequences is thus through the alignment (*Match, Deletion, Match, Insertion, Match*). This alignment is not unique, however. For example, it would also be possible to align the sequences using matches only (*Match, Match, Match, Match*), with the disadvantage that unequal symbols are matched onto each other (*B* onto *A* and *A* onto *D*), but the advantage that fewer alignment steps are required in the process. To decide which alignments are better, we define costs for the various matches, insertions, and deletions. Given these costs, the problem of finding the best sequence alignment can be solved by dynamic programming using minimum edit distance algorithms.

If we adopt a probabilistic view on the alignment problem (instead of the simpler cost perspective), we can represent the different types of mini-alignments (e.g. *Match*, *Insertion*, *Deletion*) by states of an HMM. Because such an HMM operates on pairs of sequences, we denote it with pair HMM (Durbin 1998). A simple pair HMM and an example alignment is depicted in Figure 1.

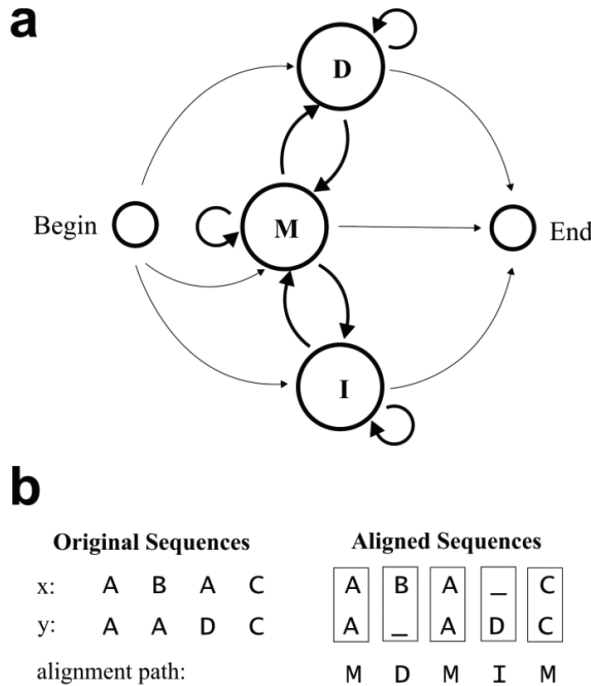


Fig. 1. A simple pair HMM for alignment (after Durbin, 1998) and alignments produced by it. (a) A pair HMM with one match state (M), one insertion state (I) and one deletion state (D) as well as a begin and an end state. (b) Two sequences x and y (left) and their alignment (right) by virtue of the state sequence: M, D, M, I, M (Match, Deletion, Match, Insertion, Match).

Using pair HMMs, the best sequence alignment is given by the most probable state path, which can be computed using an analogue to the Viterbi algorithm (Durbin 1998). We can also efficiently compute the probability that two sequences are related by *any* alignment using the forward algorithm. Independence of the scalar similarity of two sequences from any particular alignment is an advantage over the simpler dynamic programming approach outlined above, which allows only sequence comparisons by virtue of specific alignments. For example, the second and third best alignments might all be very good, which would so contribute to the similarity estimate of the pair HMM, whereas in the simple dynamic programming approach it does not. Hence, pair HMMs can provide a more robust similarity estimate than estimates based on the best alignment only.

Another advantage is that using pair HMMs we are not dependent on pre-specified costs for insertions, deletions and matches, but these parameters are embedded in emission and

transition probabilities than can be learned from data using a variant of the Baum-Welch algorithm (Rabiner 1989). Note that the Viterbi, forward, backward, and Baum-Welch algorithms for pair HMMs are derived in a straight forward manner from their standard HMM counterparts (Durbin 1998).

In the following we introduce a new pair HMM architecture that deviates from Durbin's original proposal in some significant ways. The pair HMMs contain three different types of hidden states (match, insertion, and deletion) and a continuous emission alphabet (compare Figure 1a). Durbin's model operated only on discrete observations, which is adequate for problems such as genome alignment. However, when dealing with continuous data such as birdsong, models with continuous emission probability distributions are more convenient, because they can reflect the acoustic features of song on a continuous scale.

We denote the two sequences by x_1, x_2, \dots, x_T and y_1, y_2, \dots, y_U . The pair HMM contains m deletion, m insertion, and n match states. In our notation these hidden states are distinguished by the index j : $j \in Deletion = \{1, \dots, m\}$ correspond to deletions states, $j \in Insertion = \{m + 1, \dots, 2m\}$ correspond to insertion states, $j \in Match = \{2m + 1, \dots, 2m + n\}$ correspond to match states. Furthermore, we denote

a_{ij} : transition probability from hidden state i onto hidden state j

$b_j(x_t)$: emission probability of symbol x_t given hidden state $j \in Deletion$

$b_j(y_u)$: emission probability of symbol y_u given hidden state $j \in Insertion$

$b_j(x_t, y_u)$: emission probability of symbols x_t and y_u given hidden state $j \in Match$

For our pair HMM, the recursive equations of the Viterbi algorithm are given by:

$$\forall j \in Match: \quad V_j(t, u) = b_j(x_t, y_u) \cdot \max_i (a_{ij} V_i(t-1, u-1)) \tag{1}$$

$$\forall j \in Deletion: \quad V_j(t, u) = b_j(x_t) \cdot \max_i (a_{ij} V_i(t-1, u)) \tag{2}$$

$$\forall j \in Insertion: \quad V_j(t, u) = b_j(y_u) \cdot \max_i (a_{ij} V_i(t, u-1)) \tag{3}$$

Where V is the best score (highest probability) along a single path, at times t in the first sequence and u in the second sequence, which accounts for the first t and u observations and ends in state j . In the following, we consider two realizations this pair HMM architecture.

3.1.1 Pair HMM with three states and distance metric

First we consider a very simple realization of the aforementioned architecture: a three-state pair HMM with one match, one deletion, and one insertion state. Its parameters are predetermined and not learned. Its transition matrix a is parameterized by three free parameters:

$$a = \begin{bmatrix} 1 - 2\delta - \tau & \delta & \delta \\ 1 - \varepsilon - \tau & \varepsilon & 0 \\ 1 - \varepsilon - \tau & 0 & \varepsilon \end{bmatrix} \tag{4}$$

The parameter ε is the probability of remaining in the insertion or deletion state, whereas δ is to probability of entering a deletion or insertion state from the match state and τ is the probability to transit to the end state. Transitions from deletions to insertions and vice versa are not allowed in this model (compare Figure 1a).

For the emission probability densities, we assume $b_j(x_t) = 1$ for $j \in \text{Deletion}$ and $b_j(y_u) = 1$ for $j \in \text{Insertion}$. The emission probability density $b_j(x_t, y_u)$ of the match state depends on the Euclidean distance between observations x_t and y_u as follows:

$$b_j(x_t, y_u) = k e^{-\frac{|x_t - y_u|^2}{2\sigma^2}} \quad (5)$$

where k is the trade-off factor between matches and insertions/deletions and σ is the standard deviation of the Gaussian distance metric. A good choice of k is such that the expectation of the right hand side of equation 5 over all observation pairs x_t and y_u equals 1. This reflects the assumption that random pairs x_t and y_u are about as likely to match as to form inserts or deletions. With the exception of our choice of emission probabilities $b_j(x_t, y_u)$ in the match state, this model architecture corresponds to the one presented in (Durbin 1998).

To obtain discrete-time observations x_t and y_u from birdsong, we binned the continuous-time acoustic signal. A standard practice, that we also adopted here, is to use a time-frequency representation of the signal (columns of log-power spectrograms, in our case with time-steps 5.8 ms between columns and frequency resolution 86 Hz, from 0 to 11025 Hz). The observations x_t and y_u are given by normalized columns of the spectrogram. Figure 2b depicts an alignment of two birdsongs (a tutor song and a pupil song) that was computed using the 3 state HMM.

3.1.2 Pair HMM with many states and no distance metric

Next we introduce a more powerful pair HMM with many more states and more general alignment capabilities. Songs consist of sequences of distinct syllables in which the tempo of some syllable may vary more than others. Also, when songs of two different birds have to be aligned to each other (such as tutor song and pupil song), the spectral features of their songs might be quite different so it may be difficult to find an adequate distance measure. The expansion of the simple pair HMM from the previous section solves both these problems, i.e. temporal variability can be modeled locally and songs can be aligned even if the spectral mismatch is high.

HMMs can serve as generative models of song (Kogan and Margoliash 1998). However, the pair HMM in our previous section is not a generative model of song. To adopt this capability and model single song elements by discrete states, we choose an expansion of the pair HMM as follows: we define a set of individual states for each operation (deletion, insertion, match) instead of single state each. Further, we are dealing with a full transition matrix a . Note: In the case of working solely with insertion and deletion states, such a model would be equivalent to an HMM for each song separately. We choose as model for emissions a sum of Gaussians. Such models can be trained using the Baum-Welch-algorithm.

Formally, the emission probabilities in the model are joint probability distributions over pairs of columns (for the match state) and distributions over single columns (insertion and

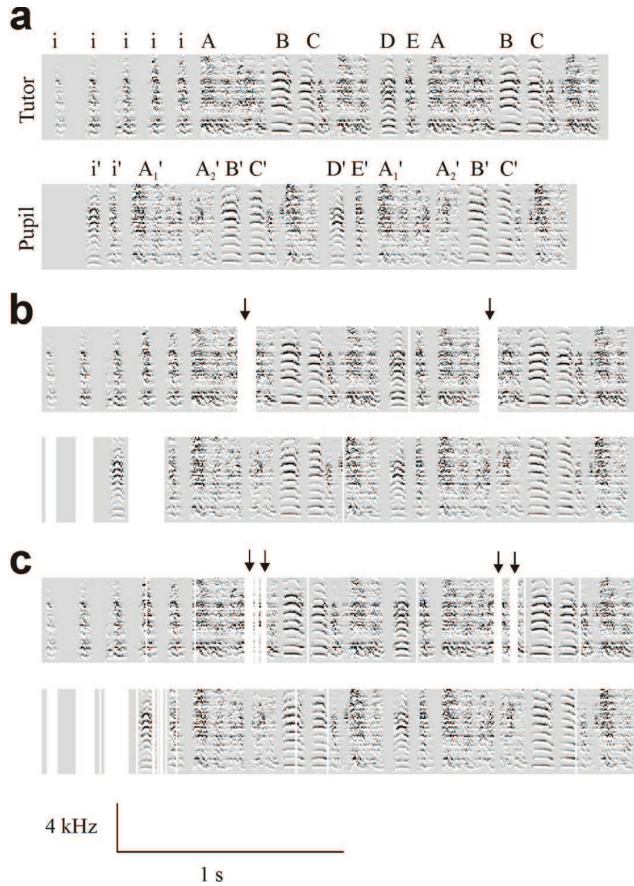


Fig. 2. Alignment of tutor song and pupil song. (a) Derivative spectrogram of tutor song and the pupil song. Visibly, the pupil copied the song of his tutor remarkably well. The main difference is Syllable A: In the tutor's song Syllable A consists of 4 parts, whereas the pupil divided that syllable into two syllables: Syllable A₁ (consisting of 3 parts) and Syllable A₂ (consisting of 1 part). (b) Alignment of the two songs by the Viterbi path using the 3 state model. Syllable A in the tutor song is split into 2 and correctly aligned to Syllables A₁ and A₂ of the pupil's song (arrows). All other syllables are correctly aligned as well. (c) Alignment of the same two songs by the Viterbi-path of a 40-state pair HMM. The model consists of $n = 20$ match states, $m = 10$ deletion states, and $m = 10$ insertion states. Emission probabilities of deletion and insertion states are modeled as the sum of two 128-dimensional Gaussians defined on the spectrum of the corresponding song. The emission probability of the match state is defined as the sum of two 256-dimensional Gaussians (defined on the spectrum of both songs). The alignment in this 40-state model is similar to the alignment in the 3-state model. However, Syllable A of the tutor song is split in 3 or 4 parts (instead of two parts), marked by the arrows. The first split accounts for the break between Syllables A₁ and A₂ in the pupil's song, whereas the other splits account for Syllable A₂ that is sung slower than the fourth part of the tutor Syllable A.

deletion), all of which can be learned from the data. In the example presented in Figure 2c, we have modeled the emissions probability densities for each state as a sum of two Gaussians with diagonal covariance matrix

$$b_j(x_t) = \sum_{k=1}^2 c_{jk} \cdot |2\pi\Sigma_{jk}|^{-1/2} \exp\left(-\frac{1}{2}(x_t - \mu_{jk})^T \Sigma_{jk}^{-1}(x_t - \mu_{jk})\right) \quad (6)$$

for deletion states and similarly for the other states. The transition probabilities a_{ij} as well as the means μ_{jk} , the weights c_{jk} , and the covariance matrix Σ_{jk} of each Gaussian were trained using the Baum-Welch algorithm on seven pairings of tutor and pupil song. The forward probabilities in our pair HMM (the probabilities $\alpha_j(t, u)$ of observing partial output sequences until times t and u and of being in hidden state j at these times) obey the recursive equations

$$\forall j \in Match: \quad \alpha_j(t, u) = b_j(x_t, y_u) \sum a_{ij} \alpha_i(t-1, u-1) \quad (7)$$

$$\forall j \in Deletion: \quad \alpha_j(t, u) = b_j(x_t) \sum a_{ij} \alpha_i(t-1, u) \quad (8)$$

$$\forall j \in Insertion: \quad \alpha_j(t, u) = b_j(y_u) \sum a_{ij} \alpha_i(t, u-1). \quad (9)$$

The backward probabilities $\beta_j(t, u)$ are calculated analogously:

$$\begin{aligned} \beta_i(t, u) = & \sum_{j \in Match} a_{ij} b_j(x_{t+1}, y_{u+1}) \beta_j(t+1, u+1) + \sum_{j \in Deletion} a_{ij} b_j(x_{t+1}) \beta_j(t+1, u) + \\ & + \sum_{j \in Insertions} a_{ij} b_j(y_{u+1}) \beta_j(t, u+1). \end{aligned} \quad (10)$$

The only parameters we have to define beforehand are the number m of deletion and insertion states, the number n of match states, and the number of Gaussians in the emission model. But there is no need to define a (spectral) distance measure; hence the spectrograms in the songs to be aligned are never directly compared to each other. The emission probabilities $b_j(x_t, y_u)$ of match states represent spectral features which are acquired during training by matching similar spectral-temporal dynamics in both songs.

3.2 Advantages and limitations of pair HMMs for birdsong alignment

As outlined above, the main use for pair HMMs is the identification of corresponding syllables in songs from either a single bird or from two different birds. However, corresponding elements can in principle also be identified by different approaches. One could for example segment the audio signal into syllables by thresholding sound amplitude, then extract suitable features and cluster all syllables based on these features. Note that in this case, we fix a priori what constitutes similarity by virtue of our choice of the features and clustering algorithm. This is not at all a problem if we are dealing with simple similarity relationships, for instance, if the songs to be aligned are highly similar and their difference can be ascribed to a source of additive noise for example. It is also not a problem if we have sufficient knowledge of the invariances of the problem, i.e. the dimensions that do not play a role in establishing similarity, because, in that case, we can select our features accordingly.

If, however, the similarity relationship is more complex and we have no a priori knowledge of it, we may prefer to learn the similarity relationship from the data using as few assumptions as possible. That is what pair HMMs can give us. If two songs can be aligned using matches, insertions, and deletions as provided by pair HMMs, we can learn the similarity relationship between them using for example the Baum Welch algorithm. In this manner our pair HMMs can deal with the case of a pupil that correctly learns the tutor song except that the pupil's song has higher pitch (or song pitch varies randomly from trial to trial). Given sufficient data, the pair HMMs would be able to detect that pitch is an invariant feature with respect to the similarity relation, it would detect that pitch essentially does not matter. Furthermore, from our pair HMM we can conveniently read out the similarity relationship from the parameters of learned emissions in match states and, in principle, also from the transitions probabilities.

Such learning does of course require computational resources. The multi-state pair HMM, outlined in 3.1.2 requires more computational resources than the simple three-state model from section 3.1.1 and many pairs of songs are needed in order to learn faithful model parameters. The computation time and memory requirements scale quadratic with the lengths of observation sequences to be aligned (compared to a linear increase in the case of normal HMMs). If we use a 100-state pair HMM and align sequences of 10^4 observation each, than we have to calculate and store $10^{10} = 10^4 \cdot 10^4 \cdot 100$ forward and backward probabilities α and β in the course of processing one sequence pair with the Baum Welch algorithm. This underlines the importance of faster approximations of the Baum Welch algorithm (e.g. Viterbi training) in the context of sequence alignment.

A more conceptual problem that we face when trying to align birdsongs, are swaps of single syllables. When learning the tutor song the pupil might change the order of syllables, e.g. the tutor sings ABCD and the pupil A'C'B'D'. Clearly, we would somehow like to have a model that can align the sequence by swapping B and C instead of using insertions and deletions. However, the Viterbi algorithm provides only one global path which in the best case either includes B and B' or C and C'. We can overcome this problem by using the idea of the Smith-Waterman algorithm (Smith and Waterman 1981) to extend the Viterbi algorithm, in order to search for local instead of global alignments. In this extension the best score of match states in Equation (1) is thresholded by a power of the free parameter θ :

$$\forall j \in Match: \quad V_j(t, u) = b_j(x_t, y_u) \cdot \max \left\{ \begin{array}{l} \max_i (a_{ij} V_i(t-1, u-1)) \\ \theta^{t+u} \end{array} \right\}. \quad (11)$$

The formula for the deletion and insertion states are similarly changed to include θ . The results are regions in which the probability $V_j(t, u)$ surpasses the threshold θ^{t+u} . In these regions we backtrack the scores in order to find the local alignments. There is the risk that the Baum-Welch algorithm will entrain the alignment of B with C' and C with B'. Such misalignment could probably be avoided by also adapting the Baum-Welch algorithm to consider possible local alignments instead of all possible global alignments.

4. Alignment of spike trains with HMMs

A common problem in neuroscience is to quantify how similar the spiking activities of two neurons are. One approach is to align the spike trains and take some alignment score as

similarity measure. To that end, Victor and Purpura (Victor and Purpura 1998) introduced a spike-train similarity measure based on the physiological hypotheses that the presence of a spike is probabilistic and that there is some jitter in spike timing. Victor and Purpura introduce two similarity measures: $D^{spike}[q]$ based on spike times and $D^{interval}[q]$ based on inter-spike intervals, where q is a free parameter. They also propose an efficient algorithm to calculate these similarity measures. In this section, we outline how Victor and Purpura's method can be implemented using pair HMMs. We discuss the advantages of a pair HMM implementation and possible generalizations. First, we briefly introduce their measures:

a) $D^{interval}[q]$

Imagine two spike trains S_1 and S_2 both represented by vectors of inter-spike intervals. Their method transforms one spike train into the other by (i) adding spike intervals, (ii) removing spike intervals and (iii) changing the duration of a spike interval by Δt . There is a unit cost for (i) and (ii), and a cost of $q * \Delta t$ for (iii). The 'distance' or dissimilarity between two spike trains is defined as the minimal cost to transform S_1 into S_2 using (i-iii). This measure can be calculated efficiently using dynamic programming.

b) $D^{spike}[q]$

This is very similar to (a). Instead of considering spike intervals, we deal with exact spike times, where the parameter q now determines the cost of shifting a spike in time. Again, spikes can also be added or removed.

(a) and (b) are computed using the same underlying efficient algorithm. Victor and Purpura use their measures to analyze spike trains elicited by a set of different stimuli. By using stimulus-dependent clustering based on their measures, they determine how well the different classes of spike trains can be separated. They find that the degree to which these classes are separable depends on the chosen value of the parameter q . They explore a range of different values for q , but find that there is no single optimal value for all cases: the choice of q needs to be made based on trial and error.

The two measures (a) and (b) can also be realized using a pair HMM. In such a pair HMM there are two states corresponding to unmatched inter-spike intervals in the two sequences, which are dealt with by adding or removing spike inter-spike intervals, respectively. Also, there is one match state in which the two inter-spike intervals are matched with associated emission probability $b_M(\Delta t) = \exp(-q\Delta t)$, where q is a free parameter and Δt the interval difference. The total costs of adding, removing, and matching intervals are encoded in the emission and transition probabilities of the pair HMM. This pair HMM can be trained on a group of spike trains that belong to a given stimulus class. Thereby, the parameter q is learned from the data. Such a simple procedure of identifying the optimal q is advantageous compared to exploring a range of possible values or even choosing a value for q a priori.

In addition to its flexibility, another advantage of the pair HMM approach is that we can now define a novel similarity measure for spike trains, based not on a particular alignment, but based on all possible alignments. Instead of defining the similarity of two spike trains as the probability of the most likely transformation, we define it as the overall probability of observing the spike trains given the model parameters. This measure corresponds to the

forward probability of the pair HMM and takes into consideration all possible transformations between the two spike trains, weighed by their probability. To make the similarity measure independent of the lengths of the spike trains, it can be defined as the average forward probability per step:

$$S(x, y) = P(x, y | \lambda) \frac{1}{\max(T, U)} \quad (12)$$

where T and U are the respective lengths of the spike trains. A spike train measure based on a probabilistic approach has also been introduced by Douwels (Dauwels, Vialatte et al. 2007). Their approach of stochastic event synchrony assumes that spike trains are equal apart from global shifts, the introduction or removal of spikes, and temporal jitter of single spikes. Such transformations can be described by a triplet of parameters describing the shift offset, the standard deviation of the timing jitter, and the percentage of spurious spikes. These parameters can be derived via cyclic maximization or expectation maximization.

5. Spike sorting with HMMs

To study the underlying neural code of complex behaviors, electrophysiology is the method of choice. For example, by implanting recording electrodes into a songbird's brain, mounted on motorized microdrives, it is possible to record from neurons while the bird engages in its normal singing behavior. The neurons spike trains are often recorded in the extracellular space using low impedance electrodes that typically pick up the activity of several neurons simultaneously (so called multi-unit activity). Correspondingly, telling the activity from the different cells apart (finding out which unit fired which spike) is an important problem, when dealing with multi unit recordings.

The identification and classification of spikes from the raw data is called spike sorting. Most spike sorting methods consist of two steps. In a first step, spike events are extracted from the raw data. In a second step these events are classified. Difficulties arise when spikes overlap on the recorded trace (arising when neurons are densely packed and fire at high rate), compare Fig. 3. These 'overlaps' are notoriously difficult to sort. The shape of such an overlap can be complex, because the number of different spikes contributing to the shape is unknown, as is the exact time delay between them.

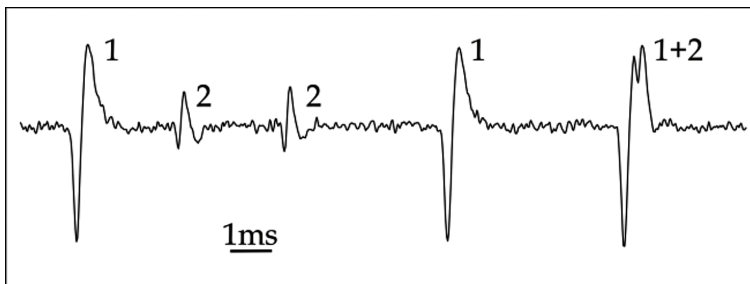


Fig. 3. Example of a multi-unit recording: Spike waveforms of two different neurons (1 and 2) are recorded on the same electrode. The rightmost waveform (1+2) represents a spike overlap.

HMMs provide a framework for addressing the spike sorting problem (Herbst, Gammeter et al. 2008). Spikes can be described by independent random variables (the hidden variables), whereas the recorded voltage is the probabilistic outcome conditional on the state of the hidden variables.

The transient nature of spikes allows us to describe neural activity by a discrete-time Markov chain with a ring structure, Fig. 4a. The first state in the ring is the resting state of the neuron, and the remaining states represent a spike. The resting state decays with fixed probability p per unit time (a free parameter), after which states along the ring are visited with unit probability (spikes are never incomplete). The recorded voltage level on the electrode is modeled as a Gaussian probability density with state-dependent mean (free model parameter) and state-independent variance (free model parameter), Fig. 4b.

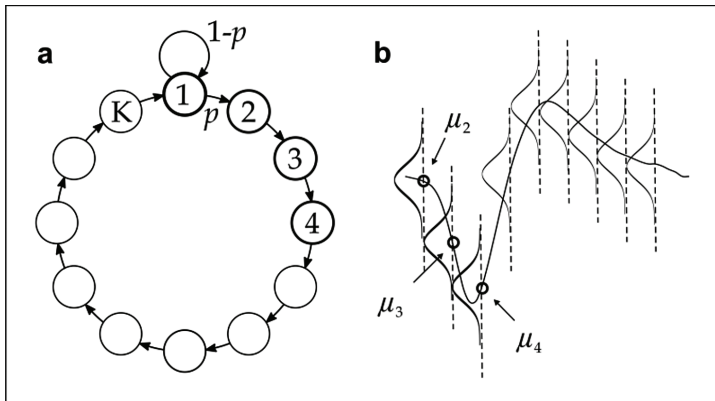


Fig. 4. (a) The state space of the hidden variable: There is a resting state that is left with probability p . Once the resting state is left, state 2 to K are visited with unit probability. (b) For each state k there is a free parameter μ_k , the mean of the Gaussian output probability. The means of the Gaussians represent the mean spike waveform.

A single ring is used to sort extracellular spikes from a single neuron, as follows: The model parameters (means, variance, spike probability) are learned by using algorithms such as the Baum-Welch algorithm. Thereafter, these parameters are used to calculate the most probable hidden state sequence with the Viterbi algorithm. The spike times are given by the time points at which the hidden variable visits state 2 (visited states start revolving around the ring).

This HMM can be extended to allow for sorting of multi-unit activity. In this extension, each neuron is modeled as an independent ring-like Markov process as in Figure 4a. The extracellular voltage signal is again modeled as a Gaussian random variable with fixed variance (free model parameter). The mean of the Gaussian is formed by summing up the state-dependent means of each neuron. Hence, each neuron contributes to the mean of the Gaussian in a linear manner. The extended framework forms a factorial hidden Markov model (fHMM) (Ghahramani and Jordan 1997).

The learning of model parameters can be performed using the expectation maximization algorithm (Baum-Welch) or using structured variational inference (Saul, Jaakkola et al. 1996). After model parameters are learned, spike sorting is performed using the Viterbi

algorithm. Using an HMM approach, spike sorting can be done in a single framework, we do not have to separate the sorting problem into separate frameworks for spike extraction and spike classification as is common using other sorting methods. The HMM operates on the recorded data from the first to the last step of the spike sorting. There is no need for pre-processing, such as spike identification, feature extraction or even manual inspection, all of which are error prone.

HMM-based spike sorting is permissive to spike overlaps. Most overlap-permissive algorithms first classify isolated spikes and then split overlapping cases to the according classes. The advantage of the HMM algorithm is that it is overlap robust both during parameter learning and during spike-sorting itself; for the model there is no difference whether spikes appear as single or overlapping events. The parameters for a neuron can even be learned in the case where the spikes are always overlapping and never appear as single events.

The HMM can easily be extended to sort spikes recorded with tetrodes. Tetrodes are electrode bundles with contacts lying within a few micrometers of each other. Depending on the distance between spiking neurons and the electrodes, the signal is picked up by several electrodes at the same time, with varying amplitude on each electrode. The signal on these electrodes is thus strongly correlated and this additional information can be incorporated into the HMM. The observed voltage level is now modeled by a multivariate Gaussian probability density with a state-dependent mean vector (free model parameters) and fixed covariance matrix (free model parameters).

Apart from spike overlaps, there are other difficulties that are often encountered in electrophysiological recordings: signal outliers (or artifacts) and nonstationarities of the data, due to bursts or electrode drift. Electrode drifts lead to a sustained change of recorded spike shape. Drifts happen slowly, they can be dealt with by updating model parameters (few iterations of the Baum-Welch algorithm) from time to time. That way the parameters are able follow the drift. Spikes produced in a very short time period, so called bursts, usually change their shape due to biophysical constraints. Spikes in bursts usually have similar spike shape as single spikes but smaller amplitudes, because not all ion channels are ready to open again. Bursting neurons can be modeled using several state rings per neuron, one ring for the first spike in a burst and the other rings for successive spikes. The introduction of new parameters can be kept to a minimum by introducing one new parameter for every ring, namely the scaling factor by which the spike template (common to all rings) is multiplied. Signal outliers should be classified as such. The HMM allows us to identify outliers both during learning and sorting as follows: the probability of an observation sequence up to time t can be calculated and can be plotted against t . From this probability trajectory one can identify the times at which the model fits the data poorly and exclude that data from learning and sorting.

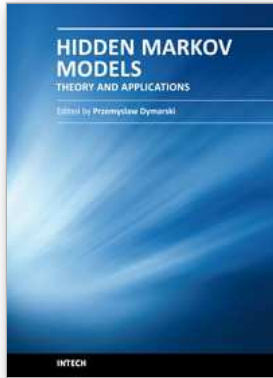
The main downside of the HMM approach for spike sorting is the computational cost involved. Assume a model with K states per ring, B rings per neuron (to model bursts) and N neurons. The number of hidden states, $(KB)^N$, grows exponentially with the number of neurons. The problem can be made tractable by imposing restrictions on the activity patterns modelled. For instance, by restricting the number of neurons that can be co-active at the same time to M , the number of hidden states can be reduced to $\binom{N}{M} (KB)^M$.

6. References

- Abeles, M., H. Bergman, et al. (1995). "Cortical activity flips among quasi-stationary states." *Proc Natl Acad Sci U S A* 92(19): 8616-8620.
- Becker, J. D., J. Honerkamp, et al. (1994). "Analysing ion channels with hidden Markov models." *Pflugers Arch* 426(3-4): 328-332.
- Brown, P., J. Cocke, et al. (1990). "A statistical approach to machine translation." *Computational linguistics* 16(2): 85.
- Camproux, A. C., F. Saunier, et al. (1996). "A hidden Markov model approach to neuron firing patterns." *Biophys J* 71(5): 2404-2412.
- Chan, A. D. and K. B. Englehart (2005). "Continuous myoelectric control for powered prostheses using hidden Markov models." *IEEE Trans Biomed Eng* 52(1): 121-124.
- Chung, S. H., J. B. Moore, et al. (1990). "Characterization of single channel currents using digital signal processing techniques based on Hidden Markov Models." *Philos Trans R Soc Lond B Biol Sci* 329(1254): 265-285.
- Danoczy, M. and R. Hahnloser (2006). "Efficient estimation of hidden state dynamics from spike trains." *Advances in Neural Information Processing Systems*, MIT Press 18: 227--234.
- Dauwels, J., F. Vialatte, et al. (2007). A Novel Measure for Synchrony and its Application to Neural Signals. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*.
- Dave, A. S. and D. Margoliash (2000). "Song replay during sleep and computational rules for sensorimotor vocal learning." *Science* 290(5492): 812-816.
- Dombeck, D. A., A. N. Khabbaz, et al. (2007). "Imaging large-scale neural activity with cellular resolution in awake, mobile mice." *Neuron* 56(1): 43-57.
- Durbin, R. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*, Cambridge Univ Pr.
- Gardner, T., F. Naef, et al. (2005). "Freedom and rules: the acquisition and reprogramming of a bird's learned song." *Science* 308(5724): 1046.
- Gat, I. and N. Tishby (1993). *Statistical Modeling of Cell Assemblies Activities in Associative Cortex of Behaving Monkeys*. *Advances in Neural Information Processing Systems 5*, [NIPS Conference], Morgan Kaufmann Publishers Inc.
- Gat, I., N. Tishby, et al. (1997). "Hidden Markov modelling of simultaneously recorded cells in the associative cortex of behaving monkeys." *Network: Computation in neural systems* 8(3): 297-322.
- Gat, I., N. Tishby, et al. (1997). "Hidden Markov modelling of simultaneously recorded cells in the associative cortex of behaving monkeys." *Network: Computation in Neural Systems* 8: 297-322.
- Ghahramani, Z. and M. I. Jordan (1997). "Factorial Hidden Markov Models." *Machine Learning* 29(2): 245-273.
- Glaze, C. and T. Troyer (2006). "Temporal structure in zebra finch song: implications for motor coding." *Journal of Neuroscience* 26(3): 991.
- Hahnloser, R. H., A. A. Kozhevnikov, et al. (2002). "An ultra-sparse code underlies the generation of neural sequences in a songbird." *Nature* 419(6902): 65-70.

- Herbst, J. A., S. Gammeter, et al. (2008). "Spike sorting with hidden Markov models." *J Neurosci Methods*.
- Immelmann, K. (1969). "Song development in the zebra finch and other estrildid finches." *Bird Vocalizations*: 61-74.
- Kemere, C., G. Santhanam, et al. (2008). "Detecting neural-state transitions using hidden Markov models for motor cortical prostheses." *J Neurophysiol* 100(4): 2441-2452.
- Kogan, J. A. and D. Margoliash (1998). "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: a comparative study." *J Acoust Soc Am* 103(4): 2185-2196.
- London, S. E. and D. F. Clayton (2008). "Functional identification of sensory mechanisms required for developmental song learning." *Nat Neurosci* 11(5): 579-586.
- Nicolelis, M., E. Fanselow, et al. (1997). "Hebb's dream: the resurgence of cell assemblies." *Neuron* 19(2): 219.
- Pawelzik, K., H. Bauer, et al. (1992). *How oscillatory neuronal responses reflect bistability and switching of the hidden assembly dynamics*, Morgan Kaufmann Publishers Inc.
- Rabiner, L. (1989). "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE* 77(2): 257-286.
- Radons, G., J. D. Becker, et al. (1994). "Analysis, classification, and coding of multielectrode spike trains with hidden Markov models." *Biol Cybern* 71(4): 359-373.
- Rainer, G. and E. Miller (2000). "Neural ensemble states in prefrontal cortex identified using a hidden Markov model with a modified EM algorithm."
- Rodriguez-Noriega, E., E. Gonzalez-Diaz, et al. (2010). "Hospital triage system for adult patients using an influenza-like illness scoring system during the 2009 pandemic--Mexico." *PLoS One* 5(5): e10658.
- Saul, L. K., T. Jaakkola, et al. (1996). "Mean Field Theory for Sigmoid Belief Networks." arxiv.org/cs.AI/9603102.
- Seidemann, E., I. Meilijson, et al. (1996). "Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task." *J Neurosci* 16(2): 752-768.
- Smith, A. C., L. M. Frank, et al. (2004). "Dynamic analysis of learning in behavioral experiments." *J Neurosci* 24(2): 447-461.
- Smith, T. F. and M. S. Waterman (1981). "Identification of common molecular subsequences." *J Mol Biol* 147(1): 195-197.
- Tchernichovski, O., P. Mitra, et al. (2001). "Dynamics of the vocal imitation process: how a zebra finch learns its song." *Science* 291(5513): 2564.
- Tchernichovski, O., F. Nottebohm, et al. (2000). "A procedure for an automated measurement of song similarity." *Animal Behaviour* 59(6): 1167-1176.
- Victor, J. and K. Purpura (1998). *Metric-space analysis of spike trains: theory, algorithms, and application*.
- Wagner, R. and M. Fischer (1974). "The string-to-string correction problem." *Journal of the ACM (JACM)* 21(1): 168-173.

Weber, A. P. and R. H. Hahnloser (2007). "Spike correlations in a songbird agree with a simple markov population model." PLoS Comput Biol 3(12): e249.



Hidden Markov Models, Theory and Applications

Edited by Dr. Przemyslaw Dymarski

ISBN 978-953-307-208-1

Hard cover, 314 pages

Publisher InTech

Published online 19, April, 2011

Published in print edition April, 2011

Hidden Markov Models (HMMs), although known for decades, have made a big career nowadays and are still in state of development. This book presents theoretical issues and a variety of HMMs applications in speech recognition and synthesis, medicine, neurosciences, computational biology, bioinformatics, seismology, environment protection and engineering. I hope that the reader will find this book useful and helpful for their own research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Blaettler Florian, Kollmorgen Sepp, Herbst Joshua and Hahnloser Richard (2011). Hidden Markov Models in the Neurosciences, Hidden Markov Models, Theory and Applications, Dr. Przemyslaw Dymarski (Ed.), ISBN: 978-953-307-208-1, InTech, Available from: <http://www.intechopen.com/books/hidden-markov-models-theory-and-applications/hidden-markov-models-in-the-neurosciences>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.