



UNIVERSITEIT
VAN
AMSTERDAM

IAS technical report IAS-UVA-05-02

Rodent behavior annotation from video

J.J. Verbeek

Intelligent Systems Laboratory Amsterdam,
University of Amsterdam
The Netherlands

In this report we describe the models with which we experimented to predict rodent behavior from video recordings. Automatic recognition of rodent behavior from video is a desirable tool in behavioral studies since it can significantly reduce the required human effort and simultaneously reduce the dependence of results on particular human observers. In our research we considered several variants of Hidden Markov Models and compared results against a simple logistic discriminant classifier that ignores the correlation of behaviors between successive frames. For a selection of four behaviors a correct classification rate around 75% is obtained.

Keywords: Classification, hidden Markov models, video analysis.

IAS

intelligent autonomous systems

Contents

1	Introduction	1
2	Description of available data	1
2.1	Data set and behaviors	1
2.2	Shape features	2
3	Behavior models for automatic annotation	3
3.1	HMM: a single frame HMM model	4
3.2	BHMM: a 25-frame batch HMM	5
3.3	KHMM: a 25-frame batch HMM with Kalman filter	5
3.4	SHMM: an HMM that models a 25-frame batch as a single feature vector	6
3.5	Logistic discriminant	6
4	Experiments	7
4.1	Performance	9
4.1.1	Performance of HMM and BHMM	9
4.1.2	Performance of MHMM and KHMM	10
4.1.3	Performance of logistic discriminant	10
4.1.4	Example of annotation performance	10
5	Conclusion	11

Intelligent Autonomous Systems
 Informatics Institute, Faculty of Science
 University of Amsterdam
 Kruislaan 403, 1098 SJ Amsterdam
 The Netherlands

Tel (fax): +31 20 525 7461 (7490)
<http://www.science.uva.nl/research/ias/>

Corresponding author:

J.J. Verbeek
 tel: +31 20 525 7550
jverbeek@science.uva.nl
<http://www.science.uva.nl/~jverbeek/>

1 Introduction

In this report we describe our research on behavior annotation of video recordings of rodents. This research is performed in cooperation with Noldus Information Technology, in an extension of the STW project AIF4997 "Tools for non-linear data analysis".

The goal is to automatically annotate a video sequence of a rodent in an arena with the behaviors exhibited by the animal throughout the video. Automatic annotation is desirable for at least two reasons. First, it will greatly reduce the cost of annotation, since manual annotation is a time consuming task currently performed by human experts. Second, an automatic annotation will not suffer from inter-observer variability experienced with human observers (different human annotators inevitably differ in their annotation). In this research we consider a "learning" approach to obtain automatic annotations. The idea is to use "examples", video segments in which the different behaviors are exhibited, to learn to discriminate between the different behaviors in other video segments. The learning approach can be formalized as a parameter estimation problem in probabilistic models, as detailed below.

The rest of this document is organized as follows. In Section 2 we describe the data that was made available by Noldus IT and used to test different methods for automatic annotation. In Section 3 we describe the different annotation methods that were tested. Then, in Section 4 we describe the experimental results obtained using these methods. We end with a conclusion in Section 5.

2 Description of available data

The video sequences are recorded from the top of the animal's arena. A foreground/background segmentation of the video frames is made, which yields a binary image for every video frame where every white pixel is part of the animal. The silhouette of the animal obtained in this manner forms the basis for automatic annotation. The recordings together with the segmentation were made available by Noldus IT. Besides the binary segmentation result, for each video several features were computed and made available by Noldus IT. In Tab. 1 we list the features which are used for behavioral annotation. In addition, features computed from the silhouette are used, these are discussed below.

2.1 Data set and behaviors

The performance tests of the annotation methods were conducted on the PT ASAPO data set. This video is recorded at 25 frames per second and has a duration of approximately 75 minutes (112494 frames). The video has been annotated using 15 different terms, listed in Tab. 2. In the same table, for each term the number of frames that are annotated by this term is also given, as well as its index number(s) in a list of terms automatically derived from the original annotation. After the apomorphine injection (at about 15m30s) the animal quickly exhibits "rear wall/grawing wall" behavior, which it remains exhibiting for about 50 minutes. Since in this time span there is relatively little variation in the behavior we did not use these video frames. Thus, about 25 minutes of the video remains which contains recordings of alternating behaviors, which is of interest for learning behavior models for automatic annotation.

In order to have enough data to learn behavior models and to assess the annotation result using the models, we selected the behaviors for which at least 4 minutes of video ($= 4 \times 60 \times 25 = 6000$ video frames) were available to perform the experiments described later in this document. These behaviors are "Rear wall / sniffing wall", "Sit", "Sniffing floor", "Sniffing wall", and "Walk". The two behaviors "Sniffing floor" and "Sniffing wall" were treated as one, since the

Feature	index
Center of gravity x	3
Center of gravity y	4
Nose point x	9
Nose point y	10
Tail point x	11
Tail point y	12
Angle of main axis	14

Table 1: Features used for behavior annotation.

position of the arena wall was not known. Note that there might be some true semantic overlap between the behaviors "Rear wall / sniffing wall" and "Sniffing wall".

In the experiments we divided the frames from each of these 4 behaviors in two disjoint sets: a training set and a test set. The training set was used to estimate the parameters of the behavior models. The test set was used to determine the quality of behavior annotation using the behavior models. This is a standard procedure in experimental machine learning research. It is important since estimating and evaluating the models from the same data can lead to overly optimistic evaluation, see also Section 4.

2.2 Shape features

In addition to the features listed in Tab. 1, we derived a set of features from the shape of the silhouette. In each frame we measured in 180 directions the distance from the center of the silhouette (as given by the center-of-gravity feature listed in Tab. 1) to the edge of the silhouette in that direction. In this way a periodic signal $s(\theta)$ is obtained, the distance s as a function of the angle θ . This signal can be described using a Fourier series:

$$s(\theta) = c + \sum_{k=1}^K a_k \cos(k\theta) + b_k \sin(k\theta). \quad (1)$$

The Fourier coefficients $c, \{a_k, b_k\}_{k=1}^K$ characterize the distance signal s .

Note that if the silhouette is rotated in the image, then the phase of the signal changes but otherwise it remains the same. Thus, the coefficients are not invariant for rotation of the silhouette. However, a rotation invariant description is desirable since it should not matter in which direction the animal is performing a certain behavior (e.g. walking), it should be recognized similarly in all directions. Two approaches can be taken to obtain a rotation invariant set of features.

1. We can use a different set of features, namely $c, \{d_k\}_{k=1}^K$ where $d_k = a_k^2 + b_k^2$. Note that in this manner the phase information is lost for every frequency, therefore also the relative phase difference between the frequencies is lost.
2. We can estimate the orientation of the animal and use this estimate to correct the Fourier coefficients for the orientation (=phase). In this manner the relative phase differences between the frequencies are preserved.

To estimate the orientation of the animal we used the angle of the main axis of the silhouette and the nose point. The main axis does not specify an orientation, since it is just a line through

Term	frames	index
Body shakes	25	2
Chewing wood chip	2025	3
Grooming snout	375	4,6
Grooming flank	175	5
Injection with Apomorphine 1.5 mg/kg s.c.	1250	7
Rear	725	1,8
Rear wall	1725	9
Rear wall/gnawing wall	73950	10
Rear wall/sniffing wall	7800	11
Sit	8850	12
Sniffing	150	13
Sniffing corner	50	14
Sniffing floor	3825	15
Sniffing wall	4300	16
Walk	7275	18

Table 2: Annotation terms and number of frames per term.

the center-of-gravity. We used the nose point to select one of the two possible directions: we project the nose point on the main axis and then chose the direction of the axis that contains the projection.

Note that although the second method preserves the most shape information, it is dependent on the main axis. If an elliptic approximation of the silhouette is near circular, then the main axis is sensitive to small changes in the foreground/background segmentation. In this way very similar shapes can yield a quite different set of features, since the phase estimate could be poor. By considering the differences in the orientation estimate between successive frames we get an impression of how often this happens. In 0.07% of the cases (82 out of 112493 frames) we found the difference to be larger than 90° , and in 0.10% of the cases (117 out of 112493 frames) the difference was larger than 45° .

We also explored a method to estimate the orientation based on a non-linear Kalman filter. Although this method gave slightly more accurate orientation estimates, its computational cost led us to use the more simple orientation estimates based on the main axis.

3 Behavior models for automatic annotation

Given the features of all video frames, the annotation task boils down to segmenting a time-series. To map the time-series to an annotation we used several probabilistic models. Let us denote the observed features of frame t by o_t and the behavior of frame t by b_t . For a sequence of observations from time t_1 to time t_2 we write $o_{t_1:t_2}$, and similarly for a sequence of behaviors.

In general, predictive classification models (e.g. for automatic annotation) can be divided in:

1. **Generative models** are models of the joint distribution over behavior and observation sequences: $p(o_{1:T}, b_{1:T})$. From the joint distribution, a conditional distribution on behavior sequences given observations $p(b_{1:T}|o_{1:T})$ can be derived, which is used for automatic annotation. Parameters are estimated so as to maximize the joint log-likelihood $\log p(o_{1:T}, b_{1:T})$

of the training data.

2. **Discriminative models** directly estimate the conditional distribution $p(b_{1:T}|o_{1:T})$. The main attraction of discriminative models is that they do not require modelling of the (potentially high-dimensional and complex) distribution of the observations, but only how the behavior depends on the observations. Parameters are estimated by maximizing the conditional log-likelihood $\log p(b_{1:T}|o_{1:T})$ of the training data.

The advantage of generative models is that it is relatively easy to estimate their parameters, and it is relatively straightforward to include dependencies between the behaviors of successive frames. For discriminative models parameter estimation is often much harder (except for a few models such as logistic discrimination), and the inclusion of temporal dependencies is harder.

Below we describe the different types of models we have explored in our experiments.

3.1 HMM: a single frame HMM model

Hidden Markov models (HMMs) [1] are a well-known models for sequential data. They are often used in speech-recognition and optical character recognition systems. In such systems the goal is to infer the sequence of words that underlies the obtained measurements. The measurements are a sequence of sound samples in speech recognition, and a sequence of images of letters in a character recognition system. In the current application, we try to infer the behavior of the animal from the features obtained from the silhouettes.

In HMMs the unknown variable b_t at time t (behavior in our case) is called the "state" at time t , the features associated with time t are called the "observation" o_t at time t . An HMM is specified by three different probability distributions:

1. Observation model $p(o_t|b_t)$: distribution over possible observations, given the state.
2. Transition model $p(b_{t+1}|b_t)$: distribution over state at time $t + 1$ given state at time t .
3. Prior distribution $p_0(b_1)$: distribution over the possible states at time $t = 1$.

Given the behavior in all the frames before time $t > 1$, the behavior in frame t is assumed to be only dependent on the behavior in frame $t - 1$. This implies that the probability of a sequence of behaviors from time $t = 1$ to time $t = T$ can be written as:

$$p(b_{1:T}) = p_0(b_1) \prod_{t=2}^T p(b_t|b_{t-1}). \quad (2)$$

Given the behaviors sequence $b_{1:T}$ the observations are assumed to be only dependent on the corresponding behavior, which implies that:

$$p(o_{1:T}|b_{1:T}) = \prod_{t=1}^T p(o_t|b_t). \quad (3)$$

We can now define the *posterior* probability on the behavior sequence given the observations:

$$p(b_{1:T}|o_{1:T}) = \frac{p(o_{1:T}|b_{1:T})p(b_{1:T})}{p(o_{1:T})}, \quad (4)$$

where

$$p(o_{1:T}) = \sum_{b_{1:T}} p(o_{1:T}|b_{1:T})p(b_{1:T}). \quad (5)$$

The number of possible state *sequences* equals the number of possible behaviors (four in our case) to power T . Thus with four different behaviors and a sequence of 10 frames we would obtain $4^{10} = 1.048.567$ possible sequences. For 50 frames, the number of sequences is about 10^{30} . For comparison, a 200GB harddrive allows storage of about 10^{23} *bits*, so about 10 million times less bits than the 10^{30} possible sequences. Therefore, the complete distribution over possible state sequences is of no practical interest, if only because it is in general not possible to store the distribution on a computer. However, using the Baum-Welch forward-backward algorithm we can compute for each time t the posterior distribution over behaviors at that time given *all* observations $p(b_t|o_{1:T})$. Furthermore, using the Viterbi algorithm we can compute the state *sequence* $b_{1:T}^*$ with maximum a-posteriori probability:

$$b_{1:T}^* = \arg \max_{b_{1:T}} p(b_{1:T}|o_{1:T}). \quad (6)$$

See [1] for a more detailed introduction to HMMs and description of the algorithms to compute the single state posterior $p(b_t|o_{1:T})$ and $b_{1:T}^*$. Both algorithms have a runtime that is linear in the length of the sequence, T , and quadratic in the number of behaviors.

In our experiments the distributions $p(o_t|b_t)$ were taken to be multivariate Gaussian distributions (with different mean and covariance matrix for each behavior). Since the video sequence can start with any of the different behaviors we have used a uniform distribution for p_0 , assigning equal probability to all behaviors in the first video frame. The transition model was estimated from the data, which can be done by computing for each behavior in what proportion of the cases it is followed by any one of the other behaviors.

Below we describe three variations on the basic HMM model that we have explored.

3.2 BHMM: a 25-frame batch HMM

The model described above assumes that the behavior can change at every single frame in the video sequence. Although this is in principle possible, the available annotation was made at a granularity of 1 sec. of video: i.e. behaviors last for an integer multiple of 25 frames.

The BHMM model is similar to the basic HMM model, but it assumes that the behavior can only change once per second. For notational convenience, we will use s_t to denote the observation corresponding to the t -th second of video, i.e. $s_t = o_{25(t-1)+1:25t}$. For a sequence of T seconds of video (thus $25T$ frames) the joint model on observations and behaviors is:

$$p(b_{1:T}, s_{1:T}) = p(b_{1:T})p(s_{1:T}|b_{1:T}), \quad (7)$$

$$p(b_{1:T}) = p_0(b_1) \prod_{t=2}^T p(b_t|b_{t-1}), \quad (8)$$

$$p(s_{1:T}|b_{1:T}) = \prod_{t=1}^T p(s_t|b_t), \quad (9)$$

$$p(s_t|b_t) = \prod_{t_2=1}^{25} p(o_{25(t-1)+t_2}|b_t) \quad (10)$$

Again we used multivariate Gaussian distributions for the observation model $p(o_t|b_t)$, a uniform prior distribution p_0 , and the transition model was estimated from the given annotation. Note that this transition model is different than that of the single frame HMM model, because for BHMM transitions on 25 frames level are modelled rather than on single frame level.

3.3 KHMM: a 25-frame batch HMM with Kalman filter

The BHMM model we just described assumes that all observations in one second of video are independent of each other given the behavior of that second, as can be seen from (10). In the

KHMM model we replace this independence assumption by a first-order linear dynamical system (LDS) assumption. That is, given the behavior a 25-frame batch of observations is assumed to be generated by a noisy first order linear dynamical system. A well-known application of First-order LDS are Kalman filters [2]. Thus, the KHMM model and the BHMM model only differ in the observation model $p(s_t|b_t)$.

Where an HMM assumes a discrete state sequence is underlying the observation sequence, an LDS assumes a continuous state sequence is underlying the observation sequence. Given the (unknown) state sequence the observations are independent, and the state sequence is assumed to be generated by a first-order Markov process. From the joint likelihood on the state and observation sequence the likelihood of the observation sequence can be obtained by integrating out the state sequence. Let $t_1 = 25(t - 1)$ and $t_2 = 25t$, then

$$p(s_t|b_t) = p(o_{t_1:t_2}|b_t) = \int p(x_{t_1:t_2}, o_{t_1:t_2}|b_t) dx_{t_1:t_2} = \int p(x_{t_1:t_2}|b_t) \prod_{\tau=t_1}^{t_2} p(o_\tau|x_\tau, b_t) dx_{t_1:t_2}. \quad (11)$$

The first-order Markov property allows us to write:

$$p(x_{t_1:t_2}) = p_0(x_{t_1}) \prod_{t=t_1+1}^{t_2} p(x_t|x_{t-1}, b), \quad (12)$$

$$p_0(x_{t_1}) = \mathcal{N}(x_{t_1}; \mu_0, \Sigma_0), \quad (13)$$

$$p(x_t|x_{t-1}, b) = \mathcal{N}(x_t; \mu_b + A_b x_{t-1}, \Sigma_b), \quad (14)$$

where we used $\mathcal{N}(x; \mu, \Sigma)$ to denote a Gaussian density on x with mean μ and covariance matrix Σ . Equation (14) tells us that x_t is Gaussian distributed with a mean that is given by a linear function of x_{t-1} . The linear function and the covariance of the Gaussian depend on the behavior. As an example, consider setting $\mu_b = 0$ and $A_b = I$, then x_t is Gaussian distributed around x_{t-1} . Thus the KHMM model allows us to model the sequence of features corresponding to each behavior as a gradually changing process, where the dynamics may depend on the particular behavior.

3.4 SHMM: an HMM that models a 25-frame batch as a single feature vector

The SHMM model again assumes a single behavioral state per second of video. Rather than assuming independence or a first-order Markov dependence between the features of the frames in a second of video, here we collect all features of all 25 frames in a second as a single (25 times higher dimensional) feature vector. The structure of the model is that of a normal HMM. The difference with the HMM model is that here the behaviors and features are represented on the granularity of seconds rather than that of the video frames. The BHMM, KHMM, and SHMM model differ only in their observation models. All three models assume a Gaussian distribution on the observations, but they make different assumptions on the dependency between the features of the different frames: independence (BHMM), first-order linear Gaussian dependence (KHMM), and arbitrary linear Gaussian dependence (SHMM).

3.5 Logistic discriminant

The last model we consider is a discriminative model. Consider the ratio of the likelihood of a given observation under two different behaviors: $p(o_t|b_t = i)/p(o_t|b_t = j)$. The logistic discriminant model assumes that the logarithm of this ratio is linear in the observation o_t :

$$\log \frac{p(o_t|b_t = i)}{p(o_t|b_t = j)} = w_{ij}^\top o_t + w_{ij0}, \quad (15)$$

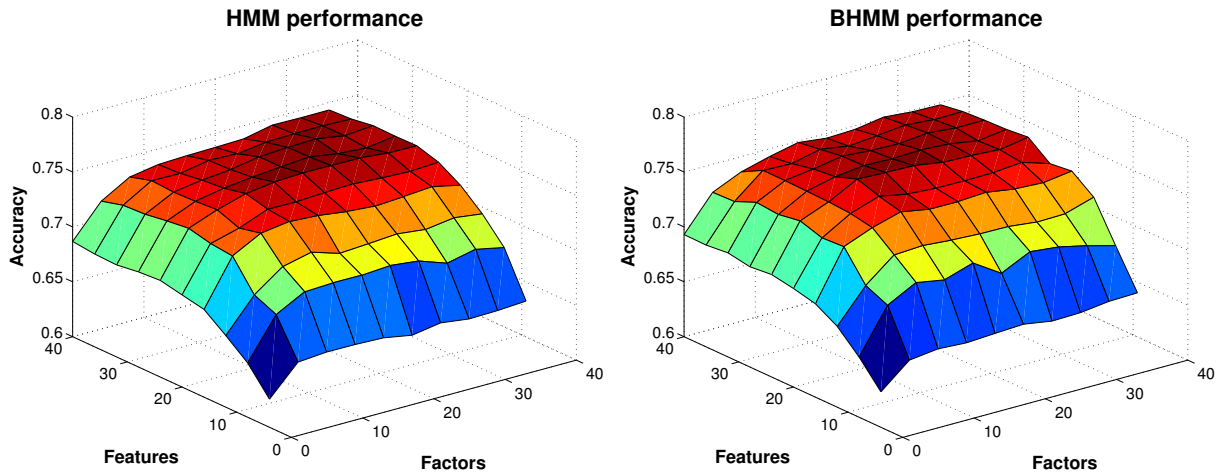


Figure 1: Annotation accuracy on test set for HMM and BHMM models for different numbers of features and numbers of factors.

where $w_{ij}^\top o_t$ denotes the inner product between the vectors w_{ij} and o_t . From this assumption it follows that the posterior probabilities on the B different behaviors given the observation can be parameterized as:

$$p(b_t = i | o_t) = \frac{\exp(w_i^\top o_t + w_{i0})}{1 + \sum_{j=1}^{B-1} \exp(w_j^\top o_t + w_{j0})} \quad (i \in \{1, \dots, B-1\}), \quad (16)$$

$$p(b_t = i | o_t) = \frac{1}{1 + \sum_{j=1}^{B-1} \exp(w_j^\top o_t + w_{j0})} \quad (i = B). \quad (17)$$

The maximum likelihood parameter estimates, maximizing the conditional log-likelihood of the correct annotation of the training data given the observations, can not be obtained in a closed-form equation for a logistic discriminant model. However, it is easy to show that the conditional log-likelihood as a function of the parameters only has a single (local and global) maximum, and therefore a gradient-ascent optimization procedure can be used to identify the optimal parameters. See e.g. [4] for a more detailed introduction to logistic discriminant models and the parameter estimation techniques.

4 Experiments

The difficulty of behavior annotation is greatly impacted by the duration length of the behaviors and by how the different behaviors follow on each other. If each behavior has a distinct and precise duration and is followed by only one other behavior, then annotation is significantly easier than when the durations of the different behaviors have great variability and each behavior can follow every other behavior.

In our experiments we used random test and train data. The random data sets were generated by segmenting the available video in segments of 25 frames (1 sec.), and randomly selecting segments for the test and train sets. To obtain a realistic sequence of behaviors we used the subsequence of the entire video in which the selected behaviors are contained. From this subsequence we randomly collected 4000 training frames for each behavior. The remaining frames (in the original order) were used as the test data. The transition models were estimated from the combined test and train data.

As observation for each frame we used:

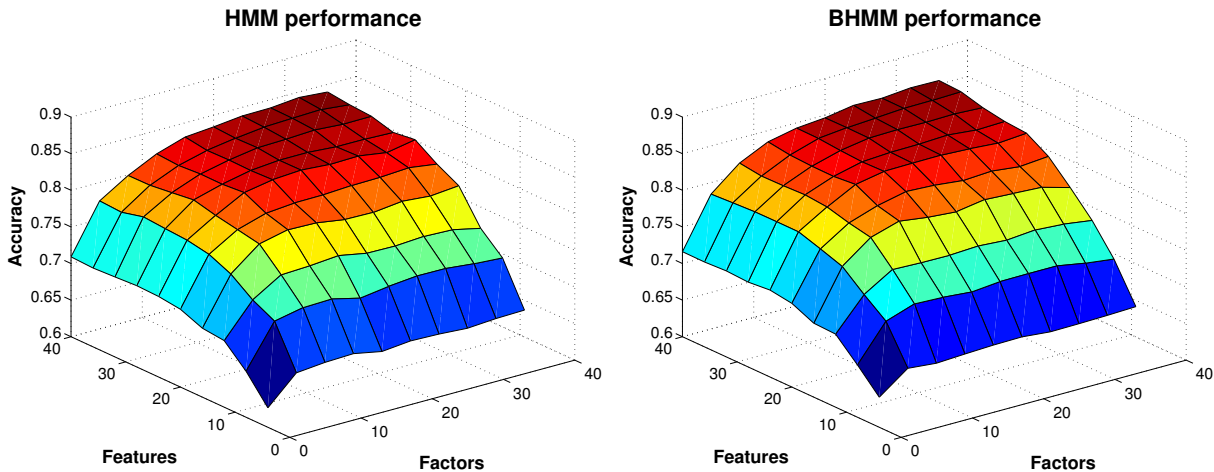


Figure 2: Annotation accuracy on train set for HMM and BHMM models for different numbers of features and numbers of factors.

1. Amount of displacement of center-of-gravity with respect to last frame.
2. Amount of orientation change with respect to last frame.
3. Distance between center-of-gravity and nose.
4. Distance between center-of-gravity and tail.
5. PCA projection of Fourier coefficients describing the oriented silhouette.

The dimensionality of the PCA (see e.g. [3] or [4] for an introduction to Principal Component Analysis) projection affects the classifier performance in several ways. First, the less features are used, the faster the classifier operates (the run-time depends linearly or quadratically on the number of features). Second, the less features are used, the less information is available to discriminate among the different classes. Third, if a very large number of features is used it is possible that overfitting occurs: the model may capture spurious data artifacts that are by chance appear to be, but are in reality not, correlated with the behaviors.

Furthermore, in the observation models there is a freedom in choosing the structure of the covariance matrix. In principle we can use a full covariance matrix Σ , but the number of parameters of the covariance matrix grows quadratically with the number of features that is used. Therefore, when using high dimensional observations there is a risk of overfitting. On the other extreme we can constrain the covariance matrix to be diagonal, which implies that all features are assumed to be independently distributed. This is a very strong assumption that probably does not hold and will lead to poor performance. As intermediate solutions we can use a Factor Analysis model, see [2], which forces all correlations between the different features to lie in a low dimensional subspace of the feature space. The covariance matrix is in this case constrained to be a sum of a low-rank matrix and a diagonal matrix: $\Sigma = WW^T + \Psi$. Here Ψ is the diagonal matrix and W is a $D \times d$ matrix. The columns of W span the d -dimensional subspace that contains the correlations, d is referred to as the number of *factors*. If $d = D$ then any full covariance matrix can be obtained. Setting $d = 0$ implements an independent feature assumption in the HMM, BHMM, and MHMM models. In the KHMM model a similar freedom exists in setting the dimensionality of the hidden state x , we also refer to this dimensionality as the number of 'factors'.

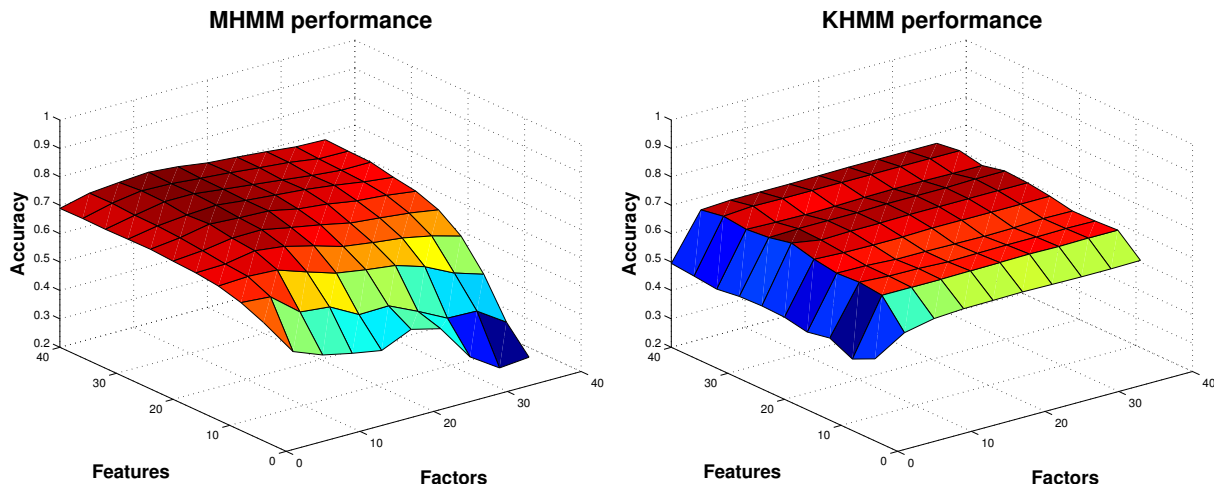


Figure 3: Annotation accuracy on test set for MHMM and KHMM models for different numbers of features and numbers of factors.

4.1 Performance

Below we report the obtained annotation accuracies for the different classifiers. We will explore the classifiers described above for different numbers of features and factors.

4.1.1 Performance of HMM and BHMM

The performance results of HMM and BHMM on the test set are given in Fig. 1. It can be seen that on average the BHMM model performs slightly better than the HMM model. However, the difference in the average performance is (almost) always smaller than one standard deviation in the results. Therefore, the two models cannot be said to yield significantly different results. Furthermore, we can see that using more features and factors is beneficial for performance until a plateau is reached starting at 16 factors and 20 features (the 4 basic features plus a 16 dimensional PCA projection of the shape features).

In Fig. 2 we show the performance results of the same models, but now we used the same data set to learn and evaluate the models. We see that generally the performance is higher in this case and that there is no plateau at which performance saturates. The fact that a higher performance level is reached in this case means that the models are capable of capturing many aspects of the training data (high performance if tested on the training data) but that the models are less good at capturing those aspects that are important for correctly classifying other data than the training data (lower performance if tested on data not used for training).

Interestingly enough, if we drop the first-order Markov dependency assumption on the behavior sequence $b_{1:T}$, and replace it by an independence assumption then the performance of the learned models drops, but not significantly.¹

We also performed experiments where the observation model $p(o_t|b_t)$ was taken to be a mixture of Gaussian (MoG) density, rather than a single Gaussian density (see e.g. [3] for an introduction to MoG models). However, experiments evaluating the performance with different numbers of mixture components did not show any improvement over the Gaussian observation models used in the experiments discussed above.

¹Thus in this we have $p(b_{1:T}) = \prod_{t=1}^T p_0(b_t)$ rather than $p(b_{1:T}) = p_0(b_1) \prod_{t=2}^T p(b_t|b_{t-1})$.

4.1.2 Performance of MHMM and KHMM

The results obtained with the MHMM and KHMM models for the test and train set are presented in Fig. 3 and Fig. 4 respectively.

For KHMM we see that poor performance is obtained when using $d = 0$ factor models or when all shape-based features are ignored. Otherwise, performance is relatively little affected by the number of factors and features. In general performance is significantly worse than that of the HMM and BHMM models.

For the MHMM model we see that the number of features has a similar impact on performance, using more factors increases performance. However, increasing the number of factors generally worsens performance on the test set, especially if a small number of features is used. On the train set, on the other hand, performance generally increases with the number of factors, unless a very small number of features is used. Clearly using a large number of factors with a large number of features leads to models that very precisely capture the correlations between features and behaviors in the train data. However, the results on the test data show that these correlations do not generalize to an independent test set of data. The difference between test and train performance of MHMM is much larger than for the other models that have been explored, which can be explained by the fact that the MHMM model is the least constrained of all models.

Also for the MHMM and KHMM models the performance is not significantly impacted by replacing the first-order Markov assumption on the behavior sequence by an independence assumption.

4.1.3 Performance of logistic discriminant

The performance of the logistic discriminant (LD) classifier as a function of the number of features is presented in Fig. 5. The results follow a similar pattern as those obtained for the different HMM models: the test error decreases as more features are used, until the performance reaches a plateau at about 20 features. The performance using the logistic discriminant model is considerably lower than that obtained using the HMM, BHMM, and MHMM models. Since the performance of the HMM models is comparable if the temporal correlation is removed, it is the fact that logistic discriminant uses linear decision boundaries in the feature space that makes it perform less well rather than its failure to capture the temporal correlation in the behavior sequence.

4.1.4 Example of annotation performance

We end the discussion of experimental results with a specific example of annotation performance obtained with the BHMM model. In this example we used $d = 12$ factors and 20 PCA features for shape (thus 20 features in total).

A test-set accuracy of 75.7% was obtained. In Fig. 6 we show the true annotation on the test set together with the annotation obtained with the BHMM model. The four behaviors are represented by colors:

1. (dark blue) "rear wall/sniffing wall"
2. (light blue) "sit"
3. (yellow) "sniffing floor" and "sniffing wall"
4. (red) "walk"

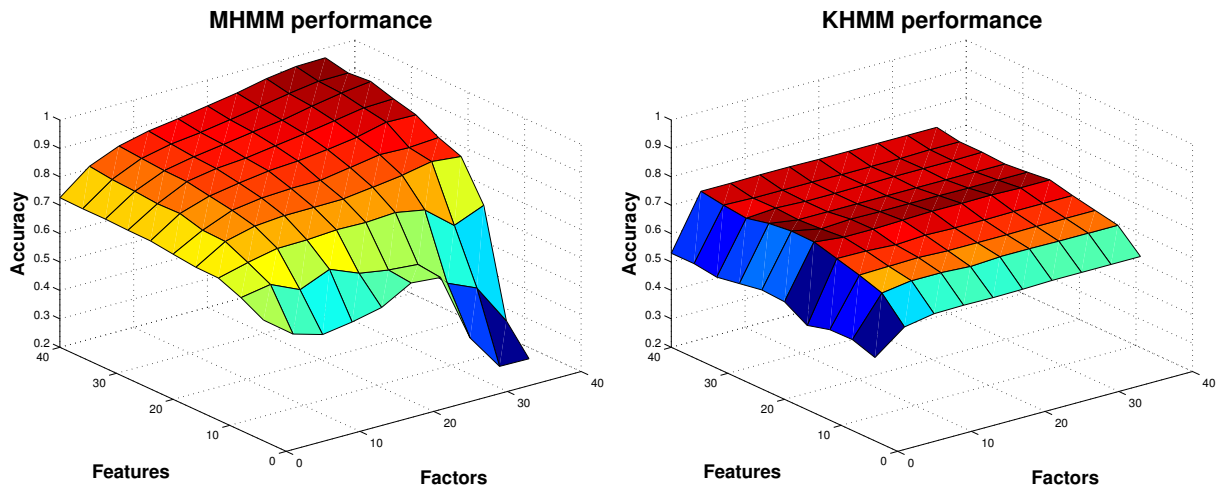


Figure 4: Annotation accuracy on train set for MHMM and KHMM models for different numbers of features and numbers of factors.

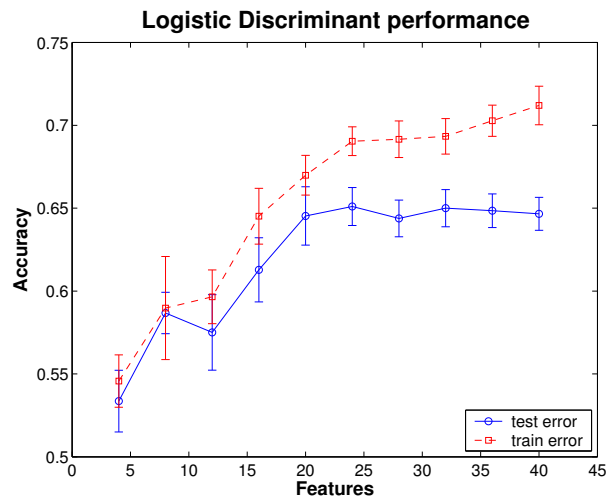


Figure 5: Annotation accuracy using logistic discriminant as function of nr. of features.

The corresponding "confusion matrix" is given in Tab. 3 in terms of frames and in Tab. 4 in terms of percentages of the frames. In these tables the behaviors are numbered as above. From the confusion matrix we see that behaviors 1 and 4 are relatively well discriminated from behaviors 2 and 3 (over 90% accuracy). Behavior 2 and 3 are less well discriminated and confused mainly with each other and behavior 4.

5 Conclusion

We have explored a collection of models for automatic annotation of video sequences of rodent behavior. The experimental results show that from the explored models, the BHMM model yields the best performance. For the annotation task involving four behaviors, a correct annotation was obtained for about 75% of the video. This is a relatively low score that should probably be improved to warrant commercial applicability of this technique. To further improve upon these results two lines of attack may be followed. First, it is possible to further explore the performance

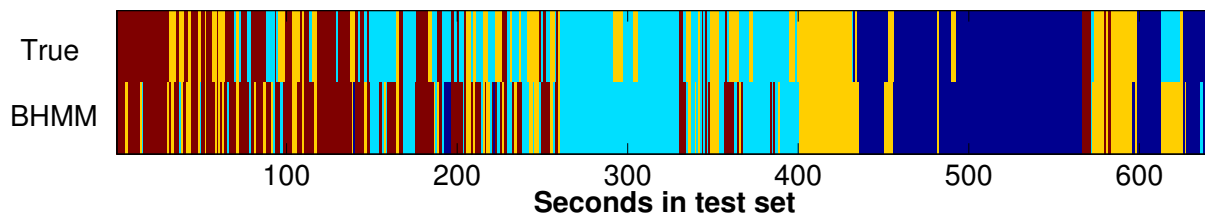


Figure 6: Annotation accuracy on train set for MHMM and KHMM models for different numbers of features and numbers of factors.

True behavior	1	2	3	4
BHMM 1	3625	125	200	0
BHMM 2	50	3000	750	75
BHMM 3	125	600	2575	250
BHMM 4	0	1125	600	2950

Table 3: The confusion matrix corresponding to the BHMM annotation.

of other models for the data obtained from the different behaviors. Second, the set of features may be extended, perhaps describing the gray level images of the animals. However, it is hard to predict beforehand which of these two possibilities will be most important to increase the level of accuracy of the automatic annotation.

In our view it is important to generate high quality data sets for further research. The data sets should be annotated by several human observers. Only in this manner it will be possible to gain insight in inter-observer variability, and to assess the performance of automatic annotation as compared to human annotation. It is also important to use a set of well defined annotation terms that do not overlap, as seemed to be the case in the annotation used in this project. Another point of attention is how exactly performance should be measured. A fixed test and train set could be made to be used in future research. However, this involves a risk since the models will be evaluated on a single data set. A better idea would be to specify and implement a method to generate test and train data sets in a randomized manner from the available data. In this manner performance can be assessed more robustly. It is important that these issues are resolved before evaluating models and feature sets, so that later results will be comparable and experiments do not have to be repeated because of a changing experimental conditions.

References

- [1] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [2] S. T. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11(2):305–345, 1999.
- [3] J. J. Verbeek. *Mixture models for clustering and dimension reduction*. PhD thesis, University of Amsterdam, 2004.
- [4] A. R. Webb. *Statistical pattern recognition*. Wiley, New-York, NY, USA, 2002.

True behavior	1	2	3	4
% of test set	23.68	30.22	25.70	20.40
BHMM 1	95.39	2.58	4.85	0.00
BHMM 2	1.32	61.86	18.18	2.29
BHMM 3	3.29	12.37	62.42	7.63
BHMM 4	0.0	23.20	14.55	90.08

Table 4: The confusion matrix corresponding to the BHMM annotation, in percentages.

Acknowledgements

This research is carried out in cooperation with Noldus Information Technology and supported by the Technology Foundation STW (project nr. AIF 4997) applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

IAS reports

This report is in the series of IAS technical reports. The series editor is Bas Terwijn (bterwijn@science.uva.nl). Within this series the following titles appeared:

J.J. Verbeek and N. Vlassis. *Semi-supervised learning with gaussian fields*. Technical Report IAS-UVA-05-01, Informatics Institute, University of Amsterdam, The Netherlands, Februari 2005.

J.M. Porta, M.T.J. Spaan, and N. Vlassis. *Value iteration for continuous-state POMDPs*. Technical Report IAS-UVA-04-04, Informatics Institute, University of Amsterdam, The Netherlands, December 2004.

W. Zajdel, A.T. Cemgil, and B.J.A. Kröse. *A hybrid graphical model for online multi-camera tracking*. Technical Report IAS-UVA-04-03, Informatics Institute, University of Amsterdam, The Netherlands, November 2004.

All IAS technical reports are available for download at the IAS website, <http://www.science.uva.nl/research/ias/publications/reports/>.