Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-7169-4

Interactive rodent behavior annotation in video using active learning

Malte Lorbach[1] · **Ronald Poppe**[1] · **Remco C. Veltkamp**[1]

Received: 31 October 2017 / Revised: 19 November 2018 / Accepted: 3 January 2019 /
Published online: 21 February 2019

Abstract

Manual annotation of rodent behaviors in video is time-consuming. By learning a classifier, we can automate the labeling process. Still, this strategy requires a sufficient number of labeled examples. Moreover, we need to train new classifiers when there is a change in the set of behaviors that we consider or in the manifestation of these behaviors in video. Consequently, there is a need for an efficient way to annotate rodent behaviors. In this paper we introduce a framework for interactive behavior annotation in video based on active learning. By putting a human in the loop, we alternate between learning and labeling. We apply the framework to three rodent behavior datasets and show that we can train accurate behavior classifiers with a strongly reduced number of labeled samples. We confirm the efficacy of the tool in a user study demonstrating that interactive annotation facilitates efficient, high-quality behavior measurements in practice.

Keywords Rat social interaction · Rodent behavior · Automated behavior recognition · Active learning · Interactive annotation

1 Introduction

The measurement of rodent behavior is of particular interest to researchers in behavioral neuroscience. Differences in the frequencies of the occurrence of specific actions and interactions can be indicative of diseases such as Huntington's [5, 32]. Traditionally, behavior measurements are performed by manually annotating hours of video recordings of rodents with pre-defined action labels, followed by an analysis of the annotations in terms of frequencies and durations [1].

With advances in computer vision and pattern recognition, we are now able to automatically annotate rodent behavior in video [12, 13, 34]. Automated annotation is typically cast as an classification task in which every frame is assigned to an action category based on features extracted from the video.

✉ Ronald Poppe
r.w.poppe@uu.nl

[1] Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands

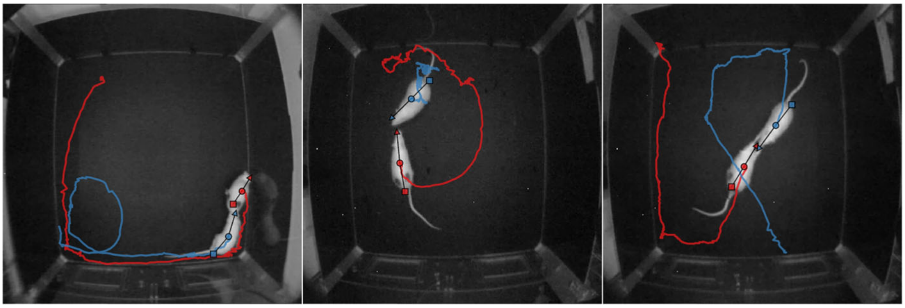


Fig. 1 Two rats recorded from top-view perspective interact in a confined arena

In contrast to human behavior, rodent actions are less structured and less goal-driven. Rodent movements are often erratic and their actions are short-lasting. The recognition of specific actions has to be based on posture and relative motion. Example frames from a rodent video are shown in Fig. 1.

Natural rodent behavior exhibits a high degree of variability across occurrences and subjects [6]. Annotation tools with pre-trained classifiers are therefore limited to a specific range of settings reflected by the training data [28]. Outside this scope, for example when new behavior classes are added or when the behaviors are performed differently due to disease progress [44], treatment or genetic background [38], the researchers need to be able to quickly train or re-train classifiers. This requires a procedure to label training examples from the new setting.

The goal addressed in this paper is to reduce the manual labeling effort by training a classifier with the *human in the loop*. We follow an active learning paradigm in which we alternate between manual labeling and classifier training. The user benefits from the option to stop labeling once the classifier is sufficiently accurate, and annotations can then be propagated throughout the remaining video material. To reduce the effort even further, the human annotator could be asked to label those examples that carry the most information for training [31]. By considering only the most informative examples, we could train a classifier that is as accurate as a classifier trained on all examples, but with significantly reduced labeling effort.

We propose an interactive annotation framework comprising three main components: selecting examples, labeling them, and training the classifier. These are the standard components in active learning applications across several domains. In this work we implement[1] and apply them to annotate rodent social behavior in two public datasets, RatSI [28] and CRIM13 [8], as well as a third validation dataset with young rats (YR). We demonstrate the merits of our approach in labeling a full dataset with a significant speed-up.

We experimentally analyze the performance and convergence properties of different selection algorithms. To find the parameters that allow us to optimally learn from rodent behavior, we first perform a series of abblation experiments. In these *offline* experiments we replace the human annotator by a dataset oracle that labels examples from previously obtained annotations. Using the oracle instead of a human allows us to test a large number of parameter settings in a short amount of time. After we have determined suitable parameters,

[1] https://gitlab.com/mlorbach/ITT

we validate the choice in a user study with human annotators. The study demonstrates the efficacy of our tool. To assure that the framework not only produces annotations for a dataset but also trains more general rodent behavior classifiers, we cross-validate the results using the related YR dataset.

The article is structured as follows. After discussing related work in Section 2, we introduce the annotation framework and implement its components specifically for rodent behavior in Section 3. We then describe the experimental setup for both offline experiments and user study in Section 4. The results are given in Section 5. We conclude in Section 6.

2 Learning to recognize rodent behavior

In order to measure rodent behavior, we need to label both type and temporal extent of every occurrence of multiple possible actions in a continuous video. This problem is typically defined as a classification task in which every video frame is assigned a discrete class label. Popular classification models for rodent behavior include neural networks [18, 35], decision trees [17, 19], and support vector machines [14, 20]. The learning algorithms used for training these classifiers are predominantly supervised [3, 8]. That is, a number of annotated training examples is used to infer the model parameters that reproduce the labels of the examples with the lowest error. Because the training data determine what can later be classified reliably, pre-trained annotation tools are limited to scenarios with a fixed set of behaviors whose definitions are universally agreed upon. In contrast, the active learning approach can allow for efficient (re-)training of classifiers in new settings. The user benefits from the extended range of settings to which the tool can be applied in return for limited manual labeling work.

The benefits of active learning are explored in a variety of multimedia applications. Typically, the goal is to make large databases accessible to humans [37] or to assists in manual annotation tasks [7, 48]. For example, vast amount of fish videos are made searchable by allowing users to define rules for interesting events [42]. Active learning can be used to efficiently obtain annotations for otherwise tedious tasks such as (re-)identifying persons in video [49], assigning meaningful attributes to objects [33], or distinguishing between normal and abnormal behavior [41]. Although active learning is also applied to annotate human activities [4, 11, 27, 47], the common assumption that the videos can be clipped to contain exactly one action is different from our application where actions occur continuously throughout longer videos.

A related way to reduce the need for labeling is to exploit information of unlabeled data in addition to a few labeled examples [45]. Such a semi-supervised approach exploits the structure in the unlabeled data to improve classification but still requires a decision as to which examples to label. Close integration of semi-supervised learning and active learning could prove a promising direction for future research.

The work closest to ours is the Janelia Automatic Animal Behavior Annotator framework (JAABA) [21]. The user of the framework labels a number of video segments while a classifier is repeatedly trained. The classifier then propagates the annotations to the unlabeled parts of the videos which in turn are judged and possibly corrected by the user. This loop continues until the user is satisfied with the generated annotations. Because the user chooses which examples to label, the quality of the final annotation depends on his selection. The criterion based on which humans compile this selection may not necessarily coincide with what is informative for the classifier. Moreover, the user has to browse through the video to find suitable segments. This takes additional time and increases the risk of missing relevant examples.

In this work we select examples automatically and actively query the user to provide a label. Because the selection is derived from the classification model, the examples are informative from the perspective of the classifier rather than the user. The interaction of human and machine is what makes the active learning paradigm powerful. We exploit the perception skills and intelligence of the human while leaving the complex, objective computations to the machine. We specifically apply this work in the context of rodent social behavior.

3 Interactive annotation for rodent social behavior

We formulate our annotation task as an active learning problem. By querying the user to label short segments in the videos, the framework learns a model of each interaction and can thus subsequently annotate the remaining parts of the videos.

We now formally introduce the active learning framework. Let us denote the set of the K behavior categories by $\mathcal{Y}$ and the domain of feature vectors by $\mathcal{X}$. The feature vector of a video frame $i(1 \leq i \leq K)$ is then $x_i \in \mathcal{X}$ and the corresponding label $y_i \in \mathcal{Y}$. We consider every frame a potential sample, although for labeling we include surrounding frames from the video to enrich the sample with context information.

The framework consists of three main components depicted in Fig. 2. There is a pool of $n - m$ unlabeled samples, $\mathcal{U} = \{x_{m+1}, \ldots, x_n\}$. Initially, all videos in the dataset are unlabeled and $m = 0$. We do not impose any specific order on the samples in the pool but we keep references to the corresponding video frames to be able to show them to the user. From $\mathcal{U}$ a sample x_i, $m < i \leq n$, is selected using sampling strategy s. The sample is then presented to an oracle, usually a human expert. The oracle provides label response y_i. The now labeled sample (x_i,y_i) is moved from $\mathcal{U}$ to the pool of labeled samples $\mathcal{L}$. From $\mathcal{L}$ the classification model $f(x)$ is learned. These steps are performed repeatedly either for a certain number of iterations or until the labeling is stopped manually. We will now describe learning and sample selection in more detail.

3.1 Learning

The task of the learning algorithm is to train the classification model with the samples that have been labeled so far. As training occurs in regular intervals, it is desirable to use an efficient learning algorithm that scales favorably with the increasing number of training samples. Besides being efficient, the classifier should also be capable of providing a confidence estimate of its predictions to be able to inform the sample selection in the next iteration.

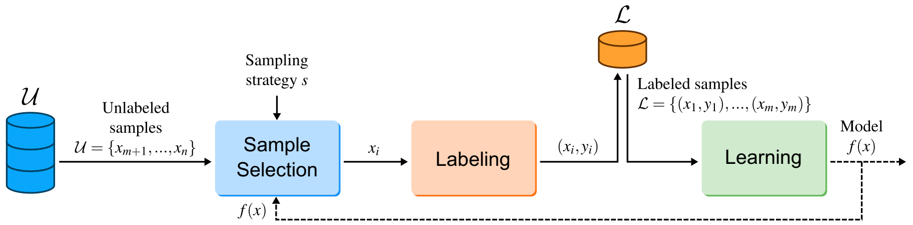


Fig. 2 Framework components. See text for details

To classify rodent interactions, we use a log-linear classification model. The model can be learned efficiently, has only one free parameter and has performed sufficiently well in previous rodent behavior experiments [28, 29]. The model allows for the calculation of a confidence score by using a suitable loss function, as we will explain shortly. Note that the log-linear model is only one possible implementation of a classification model and other types may be chosen for other applications.

We achieve multi-class classification by training multiple binary classifiers in a one-versus-all scheme similar to related work [14, 21]. Each of the $K = |\mathcal{Y}|$ classifiers distinguishes one class from all other classes and is determined by coefficients w_k and b_k with $k = 1...K$:

$$f_k(x) = w_k^{\mathsf{T}} x + b_k. \tag{1}$$

To classify a sample x_i, we evaluate all models and decide for the class label $\hat{y}_i \in \mathcal{Y}$ with the highest positive output given by

$$\hat{y}_i = \underset{k}{\operatorname{argmax}} f_k(x_i). \tag{2}$$

The K binary classifiers are trained independently of each other. Here we describe the general training procedure for one classifier $f(x)$ with binary labels $y \in \{-1,1\}$. We determine the optimal model parameters w and b by minimizing the regularized training error

$$\min_{w,b} C \sum\nolimits_{i=1}^{m} L(y_i f(x_i)) + \frac{1}{2} w^{\mathsf{T}} w \tag{3}$$

over the m labeled examples in the training set $\mathcal{L} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. The loss function $L : z \mapsto [0, \infty) \in \mathbb{R}$ and the regularization term $w^{\mathsf{T}} w$ compete for the conflicting goals of low classification error and low model complexity, respectively. This trade-off is controlled by the free parameter C. The loss function implies a cost for predicting $f(x_i)$ when the true label is y_i. We solve (3) using the LIBLINEAR library which uses Coordinate Descent for finding the minimum [15].

In order to inform the sample selection with label predictions and corresponding confidences, we derive confidence scores from the classifier output by using the logistic loss

$$L(y_i f(x_i)) = log(1 + exp(-y_i f(x_i))) \tag{4}$$

in the minimization problem in (3). With this loss function we effectively train a logistic regression model that allows us to interpret the model outputs probabilistically. First, we estimate the posterior probability that the binary classifier correctly classifies sample x_i as positive ($y_i = 1$) using the sigmoid function:

$$p(y_i = 1|x_i) = \sigma(f(x_i)) = \frac{1}{1 + exp(-f(x_i))}. \tag{5}$$

Then, the posterior probabilities from all K classifiers are normalized to a confidence score $\tilde{p}(y|x)$ whose sum over all classes is one. We will later use this confidence score to select potentially informative query samples (Section 3.2).

Before we move to the sampling strategies, we introduce one additional feature addressing the skewed distribution of behaviors in rodent data. As the learning algorithm usually puts more effort into avoiding mistakes of frequent behaviors compared to rare behaviors, the classification models for the latter become less accurate. To counter this imbalance during learning, we weigh the classes by a factor c_k, reciprocal to the relative class size as introduced previously [50]. The weight is multiplied with a sample's loss and is computed by $c_k = m/(K \cdot l(k))$, where $l(k)$ is the number of samples of class k in the labeled training set $\mathcal{L}$ (which has m samples).

For our particular classification task, the model learning is an efficient operation that takes less than one second with a standard 2.4 GHz CPU. That allows us to simply learn a new model in every training iteration instead of updating the previous model. As learning more complex classification models may take substantially longer, it may be preferred to update the model incrementally in such scenarios. Performing an effective and efficient update is a challenging task that is studied in *online learning* [10, 23].

3.2 Sample selection

We now move to the sampling strategy $s(\mathcal{U})$ which determines which sample from the unlabeled pool is selected for labeling. Let us first assume that we have no knowledge of the properties of $\mathcal{U}$, such as the label distribution. A simple approach is to select the first sample in $\mathcal{U}$, then the second, and so forth. However, given that $\mathcal{U}$ is created from videos and that labels of consecutive samples are correlated, it is desirable to spread the selection across behaviors, events and videos. We can therefore better select *random* samples.

Yet, the random selection strategy does not consider any information from already labeled samples. It is hence prone to select samples that a) are similar to previous samples and hence redundant, and b) belong to the same behavior class. The latter is true in particular for rodent behavior applications where different classes occur with different frequencies and durations. The random strategy would fill the labeled pool $\mathcal{L}$ with samples from the majority classes and neglect smaller classes. Lacking a sufficient number of training examples may cause the framework to learn inaccurate models of the less frequent behaviors.

To make better informed decisions, the sampling strategy can be guided by the classification model. Sampling then becomes a function of the unlabeled samples given the classifier: $s_f(\mathcal{U})$. This allows us to consider the informativeness of individual samples as well as the balance among behaviors. We implement a *balanced* strategy that selects two samples of each class, as predicted by the classifier, for every batch of ten queries. Because some predictions may be false, it is not guaranteed however that a perfectly uniform distribution across behaviors is achieved.

Beside balancing among behaviors, we also want to select samples that are informative with respect to learning the classification model. We first need to define what constitutes the expected information of a sample. As criterion for informativeness we utilize the classifiers' confidence in the prediction of a label. The intuition is that the sample that the classifier is least certain about is the most informative to learn from. Hence, a straightforward approach is to select the sample x^* with the lowest confidence in the highest scoring label $\hat{y}$ [26]:

$$x^* = \underset{x}{\operatorname{argmin}} \tilde{p}(\hat{y}|x). \tag{6}$$

Recall that the confidence $\tilde{p}(\hat{y}|x)$ is a real-valued number in the range [0,1] with 0 being least confident and 1 being maximally confident. The sum of the confidence values for all possible labels of one sample is 1.

Selecting the least confident sample can be a good choice for classification tasks with clearly separable data [39]. Rodent behavior however is often ambiguous [43], in particular the transitions between interactions. Querying according to the lowest confidence is likely to result in many samples being inherently difficult to label consistently for the human expert. We therefore generalize the selection criteria in (6) to select the sample that is closest to an arbitrary confidence level:

$$x^* = \underset{x}{\operatorname{argmin}} |\mathrm{CL} - \tilde{p}(\hat{y}|x)|, \tag{7}$$

where CL $\in [0, 1]^{\mathbb{R}}$ expresses the desired confidence level for the sample to be selected.

We further extend the confidence-based strategy by reintroducing some explorative abilities. Given a classification model and a pool $\mathcal{U}$, the selection criteria in (7) is deterministic. By converting the selection to a *probabilistic* sampling, we allow for more randomness. We assign each sample $x \in \mathcal{U}$ a weight ν,

$$\nu(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(\mathrm{CL} - \tilde{p}(\hat{y}|x))^2}{2\sigma^2}\right), \tag{8}$$

and then draw a sample $x^* \in \mathcal{U}$ with probability proportional to the assigned weight. For our experiments we set $\sigma = 0.025$ creating a narrow, rather conservative window that allows for randomization among behavior events but prefers samples with the desired confidence level.

4 Experiments

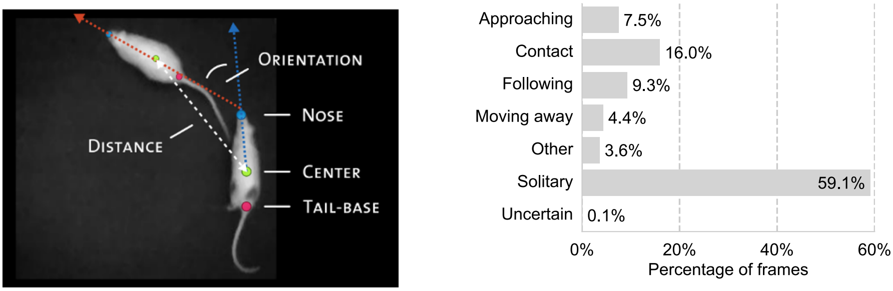
We now assess the performance of the proposed framework in the task of annotating rat social behavior videos. We evaluate the framework in two experiments. Our first goal is to determine the optimal settings of our framework, in particular regarding the sample selection strategies implemented in the previous section. This analysis is performed in offline experiments using a data oracle instead of a human annotator. The oracle responds with labels from the already annotated experiment dataset. This setup enables us to conduct a large number of experiments in a short amount of time. As there is no guarantee that the determined settings are also effective in practice when a human performs the labeling, our second experiment validates our choices in a user study. We first turn to the experiment setup.

4.1 Rodent behavior data

We use the publicly available Rat Social Interaction (RatSI) dataset [28] which contains nine videos, 15 min each, of two rats interacting in a square arena (see Fig. 1 for example frames). Every frame is labeled by one of five behavior categories: *approaching*, *contact*, *following*, *moving away* and *solitary*. In addition there is an *other* and an *uncertain* label given to undefined interactions and ambiguous frames, respectively. Note the different prior probabilities of the behaviors in Fig. 3a: in 59% of all frames the animals do not interact but perform solitary actions.

Features are derived from the tracked animal locations (center of body mass) and two additional body parts (nose and tail-base) as illustrated in Fig. 3a. The features include the animals' center- and nose-point velocities, three distances and their derivatives (center/center, nose/nose, and nose/tail-base), the relative orientation, the relative position, and the extent to which the animals overlap. Refer to Supplementary Material for details on the computation. For features whose values differ per animal (e.g. velocity), we take the mean and the absolute difference of the animal-specific values. The final feature vector consists of 17 elements.

Classification accuracy can often be improved by scaling all feature dimensions to a common range [24]. MinMax scaling normalizes the values of each dimension such that all values are in the range [− 1,1]. This type of scaling is sensitive to outliers as it relies on the minimum and maximum values per dimension. Our features are derived from tracked locations which occasionally include tracking errors. To diminish the effect of outliers,



a Distance, orientation and velocity features are derived from three tracking points per rat.

b Occurrence rates of (ground truth) behaviors, total frames = 202 500.

Fig. 3 RatSI contains videos of two rats continuously interacting with each othe

we normalize each dimension such that the 5th percentile has the value -1 and the 95th percentile the value 1.

4.1.1 CRIM13 and young rats dataset

In addition to the RatSI dataset, we make use of two other datasets. To validate the efficacy of the framework, we repeat the offline experiments on the publicly available CRIM13 dataset [8], which contains 161 videos of a residence-intruder experiment on mice. The social interactions in CRIM13 are similar to RatSI and the mice also display *approach*, *chase*, and *walk away* behavior. In addition, there is *circling* behavior in which one mouse moves in circles around the other. *Circling* occurs only sporadically and is therefore a minority class in CRIM13. As in RatSI, we treat all close-contact interactions, which we cannot distinguish reliably from the trajectory-based features [29], as one *contact* behavior. Similarly, we join actions performed by the individual animal such as *drink* and *eat* and create a *solitary* class. The label set then consists of six labels: *approach*, *circle*, *contact*, *following*, *moving away* and *solitary*.

Because in CRIM13 only the center-points of the animals are tracked, we base the computation of the orientation on the animals' motion direction. As motion direction and body orientation are not always aligned, the orientation-related features are less reliable in CRIM13. Furthermore, we cannot calculate the distances for nose- and tail-base points and therefore remove those from the feature set. We omit videos with anesthetized intruders as these lack the relevant social behavior.

For the cross-validation of the classifiers trained on RatSI we use another set of videos that are visually very similar but contain younger, and more active rats (5 weeks old, compared to 9 months in RatSI). The set comprises five videos and accurate annotations of selected segments of in total 12.5 min. As the tracking is identical to RatSI, the same feature set is used. We refer to this validation set as the Young Rats dataset (YR).

4.2 Experiment protocol

We conduct a series of learning experiments in which we evaluate three sampling strategies and different parameter settings. To monitor convergence and evaluate classification performance, we hold out 2 videos from RatSI and 95 from CRIM13 (as proposed by the authors) for the use as test set. The two videos from RatSI were chosen such that all behaviors occur sufficiently often to obtain a reliable performance measure. The remaining experiment videos form the initially unlabeled pool $\mathcal{U}$. In practice, a labeled test set is typically not available and another form of evaluation is needed. For example, additional queries can be used purely for evaluation or the human expert performs a qualitative evaluation as in [21].

We initialize the framework with one labeled sample per behavior and train the initial classifier. Although not strictly necessary, the initialization prevents several iterations of random sampling before a reasonably effective classifier can be trained. This effect is more pronounced when class distributions are unbalanced. For our experiment, initialization ensures that all experiments have the same starting point. The initialization samples are the mid-frames of randomly chosen interactions to avoid starting with an ambiguous transition. We believe that finding one example per class by scrolling through the videos is a feasible task for the human annotator in practice.

4.2.1 Querying the oracle

Because we are working with videos, a sample corresponds to a video frame. However, from a single frame the human annotator will not be able to reliably determine the performed action. Hence, we display a video clip surrounding the selected frame, which raises the question of a suitable clip duration. Short clips may not contain enough information while longer clips have a higher chance of containing more than one interaction which cannot be annotated with a single label. In pilot experiments, we found that a duration of one second is a suitable trade-off. The optimal duration may vary for other types of behavior. A video clip is constructed such that the selected frame occurs halfway through the clip. The response of the user is then assigned to all frames of the clip.

In the offline experiments, the data oracle determines its response by the majority vote over all labels in the queried clip. To prevent that a clip which contains multiple interactions is labeled by only one label, we require that the majority vote covers at least 30% of the clip. Otherwise the data oracle rejects the clip and returns the label “Uncertain”. A rejected clip accounts for one labeling iteration but the framework does not learn from rejected samples. A rejected clip cannot be queried again.

We fix the number of queries per learning experiment to 400 for RatSI and 800 for CRIM13 as these yielded sufficient examples for convergence in previous experiments. To limit the duration per experiment, we issue the queries in batches of ten. After a batch has been labeled, the model is retrained and its performance is measured. Because the learning framework may include stochastic sampling decisions, we repeat every experiment ten times using the same settings. We report the means and standard deviations for all metrics.

4.2.2 Evaluation metrics

We report the classification performance as the F1 score averaged over classes. The F1 score is the harmonic mean of the precision and recall scores. The F1 score ranges from 0, with no correct predictions, to 1 for the correct prediction of all frames. Averaging the score over classes as opposed to the total number of frames (equivalent to the ratio of correct frames) assigns equal importance to all interaction classes and prevents the score from being dominated by the most-occurring behaviors [22].

We are also interested in how the performance evolves as more training examples become available. We report the performance over time in learning curves that plot the averaged F1 score against the number of queries. To give an objective measure for comparing learning curves, we compute the area under the learning curve (AUC). The AUC combines the performance at the end of a learning experiment with the number of examples the classifier needs to reach that performance. Intuitively, we can interpret the AUC as a measure for the ability of the framework to learn efficiently. We report the area divided by the number of iterations to obtain a score in the range [0,1]. A score of 1 indicates perfect learning performance where only one learning iteration is sufficient to label all examples in the test set correctly.

4.2.3 Framework parameters

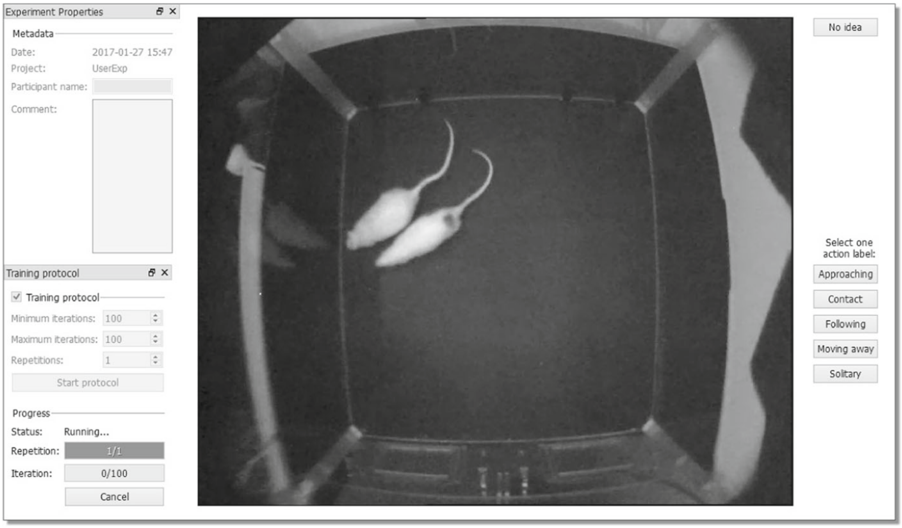
In preliminary experiments we determined suitable values for regularization parameter C and the query batch size by greedy parameter search. Best classification performance was achieved with $C = 0.1$. The query batch size has only a marginal effect on learning performance and we set it to 10 as a trade-off between learning rate and the number of retraining operations. Details are provided in the Supplementary Materials.

4.3 User study

All participants received the same written instructions about the labeling task. Next to a brief written description of the behaviors, participants were shown a short video with typical examples of each behavior. There was no time restriction for how long participants would familiarize themselves with the material (typically less than 5 min). From the five participants in total, one had no prior experience in labeling rodent interactions, and four had labeled rodent behavior before but were neither considered experts nor received professional training.

We asked the participants to label 300 clips. After every 50 clips, the labeling was paused so that participants could have a short break. Participants continued at their own pace.

The user interface of the annotation framework as depicted in Fig. 4 was specifically developed for this experiment. It shows the experiment progress, the video screen and the



Metadata
Date: 2017-01-27 15:47
Project: UserExp
Participant name:
Comment:

☑ Training protocol
Minimum iterations: 100
Maximum iterations: 100
Repetitions: 1

Progress
Status: Running...
Repetition: 1/1
Iteration: 0/100

Fig. 4 User interface of annotation framework used in user study

labeling buttons. During the experiment, the queries are automatically selected and presented by the framework. The video clips of one second are displayed in a continuous loop until the user responds by clicking one of the label buttons. Immediately after a response the next clip is played. Users have the option to reject a query if they are unsure about the correct label. Rejected queries count as one iteration but the corresponding samples are ignored for learning. Rejected samples cannot be queried again.

We also time the participants' responses. The response time is the time from the start of a video clip to the moment the user clicks on a label. We report the sum of all response times as the *total annotation time*. Scores are averaged across participants and mean and standard deviation are reported.

5 Results

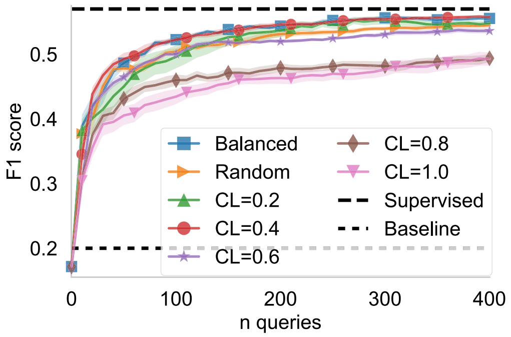
We first report the results of the offline experiments with dataset oracle on RatSI and CRIM13, followed by the user study and the cross-validation on YR.

5.1 Offline experiments

5.1.1 RatSI

We report the learning performance of our framework in Fig. 5. After 400 iterations the classifiers have converged to an accuracy that is close to the supervised classifier but using only 6.4% of the samples in the experiment set. This demonstrates the redundancy of many samples in the dataset. Comparing the performance of the different sampling strategies, we find that balanced sampling leads to better classification accuracy than random sampling. This confirms our intuition that the random strategy does not sample sufficiently from the smaller classes. If we were to proceed with the labeling, the two strategies would eventually converge to the same performance as also the random strategy will encounter rare examples. Once the training set includes sufficient examples from all classes, the learner would not benefit from the more balanced set.

Turning to the informed, confidence-based sample selection, we find that confidence level CL = 0.4 leads to the best learning performance. A confidence level of 0.2, equivalent



a Learning curves

	AUC		F1	
	mean	std	mean	std
Balanced	0.53	0.01	0.56	0.01
Random	0.51	0.01	0.54	0.01
CL=0.2	0.51	0.03	0.55	0.01
CL=0.4	0.53	0.01	0.56	0.01
CL=0.6	0.50	0.01	0.54	0.01
CL=0.8	0.46	0.02	0.49	0.02
CL=1.0	0.45	0.03	0.49	0.03

b Statistics

Fig. 5 Results with different sample selection strategies on RatSI with data oracle

to highly uncertain samples, yields lower performance. The performance also decreases for higher confidence levels due to the potentially more redundant samples.

Let us examine whether there is a link between the confidence level and the redundancy of a sample. We measure the redundancy of the selected samples by the ratio of queries for which the prediction of the sample's class matches the label given by the oracle. The intuition is that if the classifier predicts a sample correctly before being queried, the sample is less informative than a misclassified sample. Figure 6 shows the measure for the different confidence levels as well as the balanced strategy for reference. The result supports our intuition: samples with higher confidence values belong more often to the predicted class, while samples with lower confidence values are less often what the classifier predicts. Notably, below about 0.8 the confidence level is a reasonably accurate predictor as to how often the classifier is correct. This confirms that the confidence score generated by the classifier is a reliable measure for its uncertainty. A high confidence however does not guarantee correct prediction.

Although the different confidence levels lead to plausible effects in both learning and classification performance, the overall gain of using confidence-based sampling over the balanced strategy is limited. At the same time, selecting low-confidence samples interferes with the goal to balance the selection equally among all classes as more samples will be labeled differently than predicted. Therefore, choosing a confidence level involves a trade-off between a balanced but more redundant selection on one side and a random, more informative but potentially also more ambiguous selection on the other side.

5.1.2 CRIM13

On CRIM13, we use the same learning and classifier parameters as for RatSI ($C = 0.1$, CL $= 0.4$) and compare the random strategy with the balanced, confidence-weighed sampling strategy. The learning results are shown in Fig. 7.

The framework is also able to train a classifier for the CRIM13 dataset with a different feature set than for RatSI and slightly different behavior categories. Compared to RatSI we need more training examples. Although the accuracy after 800 iterations (0.48) has not converged to the supervised reference performance yet (0.52), the results demonstrate the reduction of labeling effort (800 labeled clips correspond to only 1.2% of the training set). Looking at the ratio of examples that have been labeled per class in Fig. 7b, we notice that the balanced strategy has indeed selected more minority examples than the random strategy.

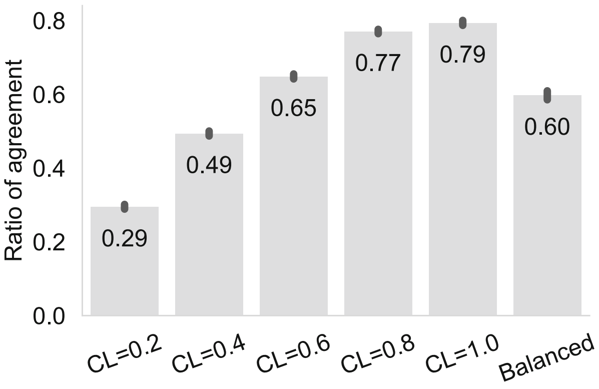
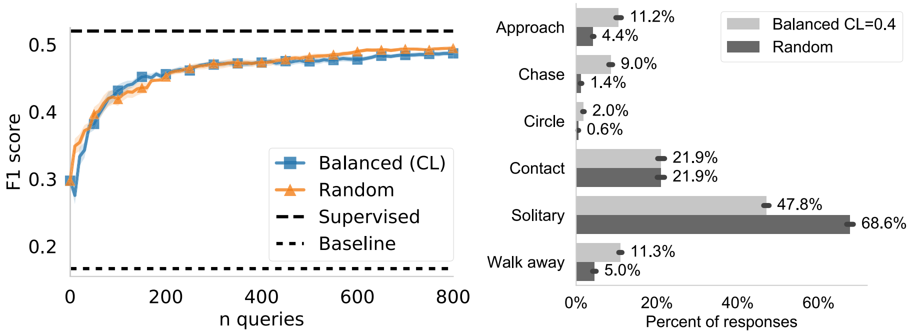


Fig. 6 Agreement between target class and label response for confidence levels (CL)



a Learning curves **b** Labeled examples after 800 iterations

Fig. 7 Learning performance on CRIM13 with data oracle

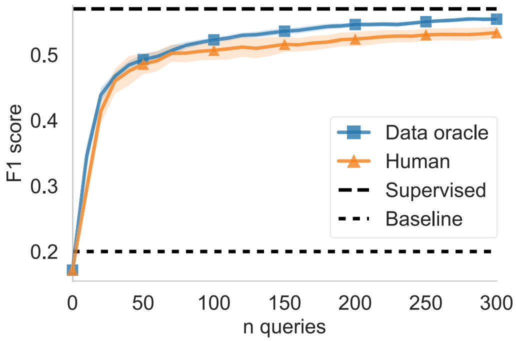
Despite this 2 to 7-fold increase in minority samples, we do not observe any substantial difference in learning performance between the two sampling strategies.

5.2 User evaluation

We now evaluate the annotation framework in practice using human annotators. In this user study we focus on the choices we made regarding the different framework settings: is a human annotator able to train an accurate classification model with the same settings?

The participants needed on average 18.7 min for labeling 300 clips excluding breaks and they rejected on average 5.7% of the queries as *uncertain*. The average time needed for labeling one minute of video using the interactive framework was 3.7 min which is comparable to the time needed in traditional, sequential annotation tasks, usually between 3 and 10 min [2, 8, 46].

In contrast to traditional labeling, the interactive framework not only obtains annotations from the user, it also trains a behavior classifier at the same time. In Fig. 8a we see that after 300 labeling iterations the classifiers are as accurate as classifiers trained by the data oracle.



	AUC		F1	
	mean	std	mean	std
Data oracle	0.52	0.01	0.55	0.01
Human	0.50	0.03	0.53	0.02

a Learning curves **b** Statistics

Fig. 8 Learning performance with human annotators on RatSI

Also the learning rates (AUC) are comparable. The results validate the framework settings that we determined with the data oracle.

The data oracle achieves a slightly higher accuracy score than the human annotators because it uses the ground truth annotations from the dataset. This yields two advantages. First, the data oracle's score does not suffer from the inter-annotator disagreement [25] between its labels and those in the validation set. Second, the expert who provided the ground truth annotations was not restricted to labeling one second clips and could thus exploit contextual information. Therefore, the labeling of the data oracle can be assumed to be more informative.

Let us take a look at the annotations generated by the trained classifiers. We first qualitatively examine the annotations in a reduced feature space. Using a t-SNE [30] projection, we reduce the features to two dimensions while preserving local neighborhood structure. This projection is shown in Fig. 9 where samples are color-coded given the ground truth labels and the automatically generated labels from one of the classifiers trained in the user study. In comparison, both sets of labels are fairly aligned on a global scale but show differences on the local scale. In the ground truth labels the behavior classes overlap substantially, whereas the generated labels are much more aligned with the local structures of the feature space and hence form relatively pure clusters. The overlap of classes emphasizes the high level of ambiguity in the ground truth labeling.

To compare the annotations quantitatively, we calculate the F1 score between each pair of annotations. Table 1 averages the results within and across groups, that is repetitions (data oracle) and study participants (human). Note that these are not accuracies in terms of the ground truth labels but agreements among the *predictions* of different classifiers. Within each group, oracle or human, the agreement is high with an average F1 score of 0.80 and 0.79, respectively. Between oracle and humans the agreement is lower with a score of 0.70. We find that the human-trained classifiers tend to predict more often *moving away* and less often *following* compared to the oracle-trained classifiers. This difference in the prior probabilities seems to be responsible for the lower agreement score, which indicates that the users and the dataset expert may have interpreted the interaction classes somewhat differently.

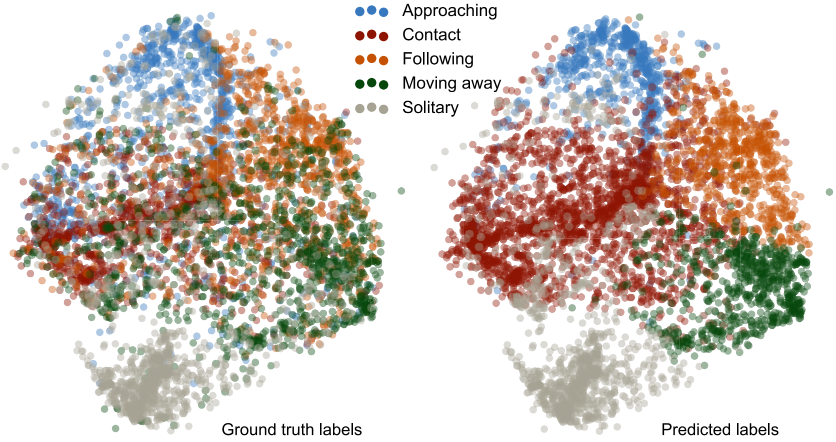


Fig. 9 t-SNE [30] visualization of 1000 random frames per behavior. Colors represent ground truth labels (left) and predictions of trained classifier (right). Best viewed in color

Table 1 Average pairwise agreement (F1 score averaged across behaviors categories) among the annotations generated by the trained classifiers

	Data oracle		Human	
	mean	std	mean	std
Data oracle	0.80	0.03	0.70	0.05
Human	–	–	0.79	0.04

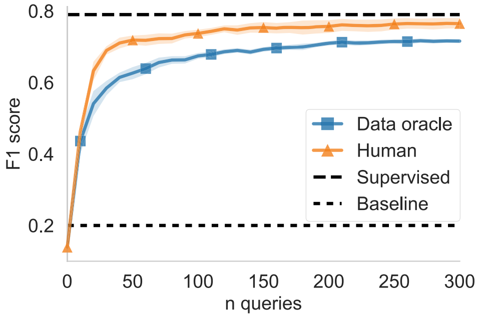
5.3 Cross-dataset validation on YR

To evaluate whether the classifiers trained by the users are applicable beyond the training videos, we validate them on the YR dataset. In addition to the overall performance of the user-trained classifiers, we also compute the learning curves using the intermediate classifiers after every labeling iteration. For reference, we include the *supervised* performance of the classifier trained and tested on YR using a 5-fold cross-validation scheme.

The results are shown in Fig. 10. The human annotators of our study outperform the data oracle in both learning rate and classification performance. The classification performance (0.76) is only slightly lower than the supervised, within-dataset reference (0.79). This difference between within and cross-dataset evaluation (0.03) is in line with our previous observations in cross-dataset experiments [28] (a difference of 0.02 using a Gaussian Mixture Model for classification).

Remarkably, the data oracle performs substantially worse. The per-class accuracies reveal that the classification performance on the *moving away* category is worse than for human annotators. As mentioned before, the human annotators have labeled more *moving away* samples than the data oracle. This difference in prior probabilities appears to result in an accuracy discrepancy on the YR dataset, whereas on RatSI it did not affect the average performance as strongly. A possible explanation is that *moving away* interactions in YR typically include clear and sudden changes in velocity. As the rats in RatSI are slower and more inert, their movements can appear more ambiguous in the one-second clips. The participant labels seem to align better with the characteristics of the young rats' behavior. This case is a good example of how different interpretations of behavior definitions affect the corresponding annotations and thus the accuracy score.

Overall, the cross-dataset result demonstrates that our annotation framework allows training rodent behavior classifiers that are not necessarily limited to one setting. In practice, this can decrease the manual effort even further as the trained classifier may be used to annotate also new videos beyond the initially considered experiment. However, concerns



a Learning curves

	AUC		F1	
	mean	std	mean	std
Data oracle	0.66	0.02	0.72	0.02
Human	0.72	0.03	0.76	0.04

b Statistics

Fig. 10 Learning performance with human annotators validated on YR

regarding the validation of the resulting annotations are as relevant as when employing a pre-trained classifier [28]. Hence, some form of evaluation is recommended, at the minimum a qualitative one as in the JAABA framework [21].

5.4 Discussion

One insight gained from the experiments is that the number of labeled samples rather than the information they carry were more important for learning. This result indicates that our sampling strategies are not yet fully effective. On the one hand, this shortcoming could be caused by an inadequate criterion for the expected information. We based this criterion on the uncertainty of the classifier. On the other hand, the selection of clips clearly relates to a qualitative level of uncertainty. Clips that were predicted with high certainty show indeed clear and prototypical interactions. Clips with low certainty show ambiguous behavior or transitions between interactions. This disagreement between human intuition and quantitative evaluation demonstrates the discrepancy between what seems informative to a human and what is actually informative for the learning algorithm.

We may improve the framework in this aspect by following a different criterion for selecting samples such as the expected increase of classification accuracy [9, 36] or the expected change in the classification model [16, 40]. Such strategies could avoid selecting samples from ambiguous regions in the feature space where most transitions are located.

Considering the trained classifiers, we found that inter-annotator disagreement affects the evaluation and the comparison between data oracle and human annotators. The disagreement is also transferred to the annotations generated by the trained classifier. As a consequence, our interactive approach to annotate behavior does not excuse us from meticulously defining the relevant behavior categories. As long as the annotator labels the behaviors consistently, the classifier will learn to consistently annotate accordingly. This restriction is a general concern of measuring behavior based on human-defined categories and therefore constrains any automated method that learns from user-labeled data.

6 Conclusion

We have presented an interactive framework that enables researchers to quickly annotate rodent behavior videos with a strongly reduced amount of work. The key to the reduced effort is to put the human in the annotation loop. With this active learning setup, we can avoid sequential and redundant labeling of similar samples and decide to stop labeling once the classification is sufficiently accurate. From our offline experiments with a dataset oracle, we conclude that the dominant factor for reducing labeling effort is the non-sequential labeling with a stopping criterion. Although balancing samples across interaction classes and selecting more uncertain samples slightly improve the learning rate, the impact on the reduction is modest. We confirmed these results in a user study in which human annotators were able to train an accurate classifier in less than 30 min which then enabled them to annotate the remaining videos with no additional manual effort.

The framework outputs a trained behavior classifier that can be used to annotate even more videos from similar experimental settings without additional manual labeling as we have validated in a cross-dataset experiment. As such, the speed-up in annotation can even be larger. Nonetheless, an interactively trained classifier is unlikely to be more general than a classifier trained on a large training set. An adequate evaluation is essential.

A number of issues are to be addressed in the future in order to expand the functionality of the framework. A common problem of the active learning approach is the question of how to measure the classification performance during labeling. Because the labeled training samples are selected by the algorithm, they are not independent and can not be used for evaluation. A potential solution is to query additional samples randomly purely for evaluation purposes or to perform a qualitative assessment.

Moreover, the framework could be extended to provide more flexibility for dealing with behavior transitions and behaviors of different durations. For instance, letting the user continue watching after the selected clip could allow the division of the clip at a transition and thus the annotation of each part separately. These extensions to the framework are solely in the user interface and do not require any modifications in the learning algorithm.

In conclusion, we have shown that interactive labeling can be used to annotate rodent behavior with strongly reduced manual effort. We are confident that extensions to the framework will allow for the annotation of an even larger range of relevant rodent behaviors in video and for studying previous data in new light with limited manual work.

Acknowledgements We thank all participants of the user study for their efforts. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for parts of the experiments.

References

1. Altmann J (1974) Observational study of behavior: Sampling methods. Behaviour 49(3):227–266
2. Anderson DJ, Perona P (2014) Toward a science of computational ethology. Neuron 84(1):18–31
3. Arakawa T, Tanave A, Ikeuchi S, Takahashi A, Kakihara S, Kimura S, Sugimoto H, Asada N, Shiroishi T, Tomihara K, Tsuchiya T, Koide T (2014) A male-specific QTL for social interaction behavior in mice mapped with automated pattern detection by a hidden Markov model incorporated into newly developed freeware. J Neurosci Methods 234:127–134
4. Bandla S, Grauman K (2013) Active learning of an action detector from untrimmed videos. In: Proc conf computer vision (ICCV), pp 1833–1840
5. Beglinger LJ, O'Rourke JJF, Wang C, Langbehn DR, Duff K, Paulsen JS (2010) Earliest functional declines in Huntington disease. Psychiatry Res 178(2):414–418
6. Benjamini Y, Lipkind D, Horev G, Fonio E, Kafkafi N, Golani I (2010) Ten ways to improve the quality of descriptions of whole-animal movement. Neurosci Biobehav Rev 34(8):1351–1365
7. Bianco S, Ciocca G, Napoletano P, Schettini R (2015) An interactive tool for manual, semi-automatic and automatic video annotation. Comput Vis Image Underst 131:88–99
8. Burgos-Artizzu XP, Dollár P, Lin D, Anderson DJ, Perona P (2012) Social behavior recognition in continuous video. In: Proc conf computer vision and pattern recognition (CVPR), pp 1322–1329
9. Cohn DA, Ghahramani Z, Jordan MI, Cohn D (1996) Active learning with statistical models. J Artif Intell Res 4:129–145
10. Crammer K, Dekel O, Keshet J, Shalev-Shwartz S, Singer Y (2006) Online passive-aggressive algorithms. J Mach Learn Res 7(Mar):551–585
11. De Rosa R, Gori I, Cuzzolin F, Cesa-Bianchi N (2017) Active incremental recognition of human activities in a streaming context. Pattern Recogn Lett 99:48–56
12. Dell AI, Bender JA, Branson K, Couzin ID, de Polavieja GG, Noldus LPJJ, Pérez-Escudero A, Perona P, Straw AD, Wikelski M, Brose U (2014) Automated image-based tracking and its application in ecology. Trends Ecol Evol 29(7):417–428

13. Egnor SER, Branson K (2016) Computational analysis of behavior. Annu Rev Neurosci 39(1):217–236
14. Eyjolfsdottir E, Branson S, Burgos-Artizzu XP, Hoopfer ED, Schor J, Anderson DJ, Perona P (2014) Detecting social actions of fruit flies. In: Proc conf computer vision (ECCV), pp 772–787
15. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: A library for large linear classification. J Mach Learn Res 9:1871–1874
16. Freytag A, Rodner E, Denzler J (2014) Selecting influential examples: Active learning with expected model output changes. In: Proc conf computer vision (ECCV), pp 562–577
17. Giancardo L, Sona D, Huang H, Sannino S, Managò F, Scheggia D, Papaleo F, Murino V (2013) Automatic visual tracking and social behaviour analysis with multiple mice. PLoS ONE 8(9):E74,557
18. Heeren DJ, Cools AR (2000) Classifying postures of freely moving rodents with the help of fourier descriptors and a neural network. Behav Res Methods Instrum Comput 32(1):56–62
19. Hong W, Kennedy A, Burgos-Artizzu XP, Zelikowsky M, Navonne SG, Perona P, Anderson D J (2015) Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. Proc Natl Acad Sci 112(38):E5351–E5360
20. Jhuang H, Garrote E, Yu X, Khilnani V, Poggio T, Steele AD, Serre T (2010) Automated home-cage behavioural phenotyping of mice. Nat Commun 1(6):1–9
21. Kabra M, Robie AA, Rivera-Alba M, Branson S, Branson K (2012) JAABA: Interactive machine learning for automatic annotation of animal behavior. Nat Methods 10(1):64–67
22. Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: One-sided selection. In: Proc conf machine learning (ICML), vol 97, pp 179–186
23. Laskov P, Gehl C, Krüger S, Müller KR (2006) Incremental support vector learning: Analysis, implementation and applications. J Mach Learn Res 7:1909–1936
24. Lecun Y, Bottou L, Orr G, Müller K (1998) Efficient BackProp. In: Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, vol= 1524, pp 9–50 Springer Verlag
25. Levitis DA, Lidicker Jr WZ, Freund G (2009) Behavioural biologists do not agree on what constitutes behaviour. Anim Behav 78(1):103–110
26. Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proc conf research and development in information retrieval, pp 3–12
27. Liu X, Zhang J (2011) Active learning for human action recognition with Gaussian Processes. In: Proc conf image processing (ICIP), pp 3253–3256
28. Lorbach M, Kyriakou EI, Poppe R, van Dam EA, Noldus LPJJ, Veltkamp RC (2017) Learning to recognize rat social behavior: Novel dataset and cross-dataset application. Journal of Neuroscience Methods
29. Lorbach M, Poppe R, van Dam EA, Noldus LPJJ, Veltkamp RC (2015) Automated recognition of social behavior in rats: The role of feature quality. In: Proc conf image analysis and processing, pp 565–574
30. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
31. MacKay DJC (1992) Information-based objective functions for active data selection. Neural Comput 4(4):590–604
32. Menalled LB, Chesselet MF (2002) Mouse models of Huntington's disease. Trends Pharmacol Sci 23(1):32–39
33. Parikh D, Grauman K (2011) Interactively building a discriminative vocabulary of nameable attributes. In: Proc Conf computer vision and pattern recognition (CVPR), pp 1681–1688
34. Robie AA, Seagraves KM, Egnor SER, Branson K (2017) Machine vision methods for analyzing social interactions. J Exp Biol 220(1):25–34
35. Rousseau JBI, Van Lochem PBA, Gispen WH, Spruijt BM (2000) Classification of rat behavior with an image-processing method and a neural network. Behav Res Methods Instrum Comput 32(1):63–71
36. Roy N, McCallum A (2001) Toward optimal active learning through sampling estimation of error reduction. In: Proc Conf Machine Learning (ICML), pp 441–448
37. Safadi B, Quénot G (2012) Active learning with multiple classifiers for multimedia indexing. Multimed Tools Appl 60(2):403–417
38. Schneider J, Levine JD (2014) Automated identification of social interaction criteria in Drosophila melanogaster. Biol Lett 10(10):E20140,749
39. Settles B (2011) From theories to queries: Active learning in practice. In: Proc workshop on active learning and experimental design, pp 1–18
40. Settles B, Craven M, Ray S (2008) Multiple-instance active learning. In: Advances in neural information processing systems (NIPS), pp 1289–1296
41. Sillito RR, Fisher RB (2008) Semi-supervised learning for anomalous trajectory detection. In: Proc Conf British machine vision conference (BMVC), pp 1031–10310

42. Spampinato C, Beauxis-Aussalet E, Palazzo S, Beyan C, van Ossenbruggen J, He J, Boom B, Huang X (2014) A rule-based event detection system for real-life underwater domain. Mach Vis Appl 25(1):99–117
43. Spruijt BM, Peters SM, de Heer RC, Pothuizen HH, van der Harst JE (2014) Reproducibility and relevance of future behavioral sciences should benefit from a cross fertilization of past recommendations and today's technology: "Back to the future". J Neurosci Methods 234:2–12
44. Steele AD, Jackson WS, King OD, Lindquist S (2007) The power of automated high-resolution behavior analysis revealed by its application to mouse models of Huntington's and prion diseases. Proc National Academy of Sciences 104(6):1983–1988
45. Tanha J, Someren MV, de Bakker M, Bouteny W, Shamoun-Baranesy J, Afsarmanesh H (2012) Multi-class semi-supervised learning for animal behavior recognition from accelerometer data. In: Proc Conf tools with artificial intelligence (ICTAI), vol 1, pp 690–697
46. van Dam EA, van der Harst JE, ter Braak CJF, Tegelenbosch RAJ, Spruijt BM, Noldus LPJJ (2013) An automated system for the recognition of various specific rat behaviours. J Neurosci Methods 218(2):214–224
47. Vijayanarasimhan S, Jain P, Grauman K (2010) Far-sighted active learning on a budget for image and video recognition. In: Proc conf computer vision and pattern recognition (CVPR), pp 3035–3042
48. Wang M, Ni B, Hua XS, Chua TS (2012) Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. ACM Comput Surv 44(4):25:1–25:24
49. Yan R, Yang J, Hauptmann A (2003) Automatically labeling video data using multi -class active learning. In: Proc conf computer vision (ICCV), pp 516–523
50. Zadrozny B, Langford J, Abe N (2003) Cost-sensitive learning by cost-proportionate example weighting. In: Proc conf data mining (ICDM), pp 435–442

Malte Lorbach has received his M.Sc. in Computer Science from the Technical University of Berlin, Germany. He is currently pursuing his Ph.D. degree at Utrecht University, The Netherlands. His current research interest is the objective measurement of animal behavior in video. In particular, he investigates the recognition of specific behavior categories using Computer Vision and Machine Learning. Part of his research was conducted within the European Marie Curie Initial Training Network "PhenoRat".

Ronald Poppe received a Ph.D. from the University of Twente, the Netherlands. He is currently an assistant professor at Utrecht University. His research interests include the automated analysis and understanding of natural, interactive behavior from videos, and the application in real-life settings. In 2012 and 2013, he received the most cited paper award from the "Image and Vision Computing" journal, published by Elsevier.

Remco C. Veltkamp is full professor of Multimedia at Utrecht University, The Netherlands. His research interests are the analysis of, and interaction with, images, video, music, and 3D objects, in particular the algorithmic and experimental aspects. He is coordinator of the Utrecht Center for Game Research, www.gameresearch.nl.