Mouse Behavior Recognition with The Wisdom of Crowd

by

Yuzhao Ni

B.Eng., National University of Singapore (2010)

Submitted to the School of Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author ....
School of Engineering
May 10, 2013

Certified by ....
Tomaso Poggio
Professor
Thesis Supervisor

Accepted by ....
Nicolas G. Hadjiconstantinou
Director, Computation for Design and Optimization

Mouse Behavior Recognition with The Wisdom of Crowd

by

Yuzhao Ni

Submitted to the School of Engineering
on May 10, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

Abstract

In this thesis, we designed and implemented a crowdsourcing system to annotate mouse behaviors in videos; this involves the development of a novel clip-based video labeling tools, that is more efficient than traditional labeling tools in crowdsourcing platform, as well as the design of probabilistic inference algorithms that predict the true labels and the workers' expertise from multiple workers' responses. Our algorithms are shown to perform better than majority vote heuristic. We also carried out extensive experiments to determine the effectiveness of our labeling tool, inference algorithms and the overall system.

Thesis Supervisor: Tomaso Poggio
Title: Professor

Acknowledgments

When Tommy approved my thesis and told me "You did a nice job" a few days ago, I suddenly realized that I would soon conclude my graduate life in MIT. I would like to thank all the people in the Center for Biological & Computational Learning(CBCL) lab, because, in the past two year, I truly enjoy every moment here with you. Especially, I would love to thank Tommy for his supervision and trust in the past two years. Most things would not be possible without his support. I appreciate it more than I can say.

Besides, I would like to thank Charles Frogner for helping me with the research and thesis, and training my teamwork skills. Without his help, most of the work in this thesis would not be possible. He is also the one who uncovers the "secret" undergraduate life in Harvard and anecdotes of Mark Zuckerberg.

I am fortunate to have Youssef Mroueh and Guillermo Canas as supportive friends in the past two years. Their intelligence and research attitude have greatly impressed me. They are also great lunch and dinner partners.

I also thank Gadi Geiger for his help and wise reminders. He is always the most appropriate person to talk to when I feel in trouble. He is also the best chef and coffee master. Chatting with him every day is more enjoyable than ever.

I also would like to thank Leyla Isik for introducing me to the politics of U.S.A., Jim Mutch for coaching me in pool game, Chun-kai Wang for introducing me to snowboarding, and Kathleen Sullivan for always bring us amazing food/desserts. Besides, I want to thank all my current labmates, Andrea Tacchetti, Stephen Voinea, Joel Leibo, Fabio Anselmi, Ethan Meyersfor, Carlo Ciliberto, Owen Lewis, and Yena Han.

Meanwhile, I also thank people I have collaborated with outside the lab over the past two years: Arlene Ducao for creating amazing technical demonstration, Jeremy Scott and Rahul Rajagopalan for winning the MIT Mobile App Competition. They make my life in MIT more exciting than ever.

Last but not least, I never forget to thank my parents for their endless love and mental support, my sister for clearing doubts in my life. I simply cannot imagine life

without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Detecting and classifying animal behavior from video is one of the most interesting challenges facing computer vision researchers. Recent study relies on developing state-of-the-art action recognition algorithms and applying them to animal behavior recognition. Although many automatic systems have already demonstrated some successful results in recognizing the home-cage mouse, they still perform much poorer than human annotators. This motivate us to search for alternative human-based solutions. In this thesis, we explore the method of mice behavior recognition using the crowdsourcing algorithms. In general, we would like to answer the following two questions:

1. How can we build an efficient online behaviorial annotation tool?

2. How can we infer the groundtruth if the results we get from online workers are noisy?

The chapter gives a general introduction to the problem of mice behavior recognition and our solution using the crowdsourcing system. It also covers some recent work on the topic of behavior recognition, online video annotation, and crowdsourcing algorithms.

1.1 Overview

Mouse behavioral recognition plays an important role in comprehensive phenotypic analysis on both small scale characterization of single gene mutants and the large scale study of the entire mouse genome[10]. Traditionally, manual annotation is frequently used to provide accurate behaviorial labels. However, this approach is very expensive and slow. Recently, thanks to the advances in computer vision and machine learning, robust systems can be developed to recognize objects[18, 63] and human actions[60]. The use of vision-based approaches is already bearing fruit for the automated tracking [57, 9, 73] and recognition of behaviors in insects[38, 32] and animals[11, 40, 40]. More recently, a few computer vision systems for the recognition of mice behaviors have been developed, including a commercial system (CleverSys, Inc) and several prototypes from academic groups[46, 70]. Notably, base on a computational model of motion processing in the primate visual cortex [27, 28], H. Jhuang, at el.[29, 31] develop a trainable, general-purpose, automated and potentially high-throughput system for the behavioral analysis of mice in their home-cage. They also provide a very large database of manually annotated video sequences of single-mouse behaviors. Besides, X. Burgos-Artizzu, at el.[69] propose a method for the automatic segmentation and classification of social "actions" in continuous multiple-mice video, where a novel trajectory features are used to improve the performance. Nevertheless, these automatic algorithms still perform poorer than human annotators. This motivate us to search for alternative human-based solutions.

One popular approach is to make use of the vast human resources on the Internet. Crowdsourcing, the act of outsourcing work to a large crowd of workers, is rapidly changing the way data are collected. Example projects such as the ESP game[66, 65], the Listen game[17], Soylent Grid[55], Purposive Hidden Object Game[45], and reCAPTCHA[41] have demonstrated the possibility of harnessing human resources to solve different machine learning problems. While these methods use clever schemes to obtain data from humans for free, a more direct approach is to hire annotators online. Recent web tools such as Amazon's Mechanical Turk[1] provide ideal solutions for high-speed, low cost labeling of massive data. With Mechanical Turk, it is possible to assign annotation jobs to hundreds, even thousands, of computer-literate workers and get results back in a matter of hours.

Due to the distributed and anonymous nature of these tools, interesting theoretical and practical challenges arise. For example, user-friendly web interface need to be built to ensure online workers are properly instructed and motivated to do high quality work. Moreover, even we have perfect user interface, the quality of labels obtained from annotators still varies. Some workers provide random or bad quality labels in the hope that they will go unnoticed. Even without spammers, annotators with different expertise can give responses with various accuracies. The standard solution to the problem of "noisy" labels is to assign the same labeling task to many different annotators, in the hope that at least a few of them will provide high quality labels or that a consensus emerges from a great number of labels.

The above challenges become even difficult when the video annotation is concerned. Some of the challenges includes: (1) building web video annotation tools is much harder than, for example, designing image labeling tools; (2) video labels are not independent from each other, since they arise from continuous video sequence. This additional dependency makes the analysis even more complex.

To address these difficulties, we present our study of efficiently crowdsourcing mice behavior annotation task. Our effort includes two video behaviorial annotation tools and novel algorithms to aggregate the behaviorial labels for long videos.

1.2 Mice behavior recognition task and datasets

We are interested in the mice behavior recognition task explained in H. Jhuang and T. Poggio [29, 31, 30, 58]. Basically, the task asks workers to identify 8 behaviors of mice in videos. The videos contain singly housed mice from an angle perpendicular to the side of the cage as shown in Figure 1-1.

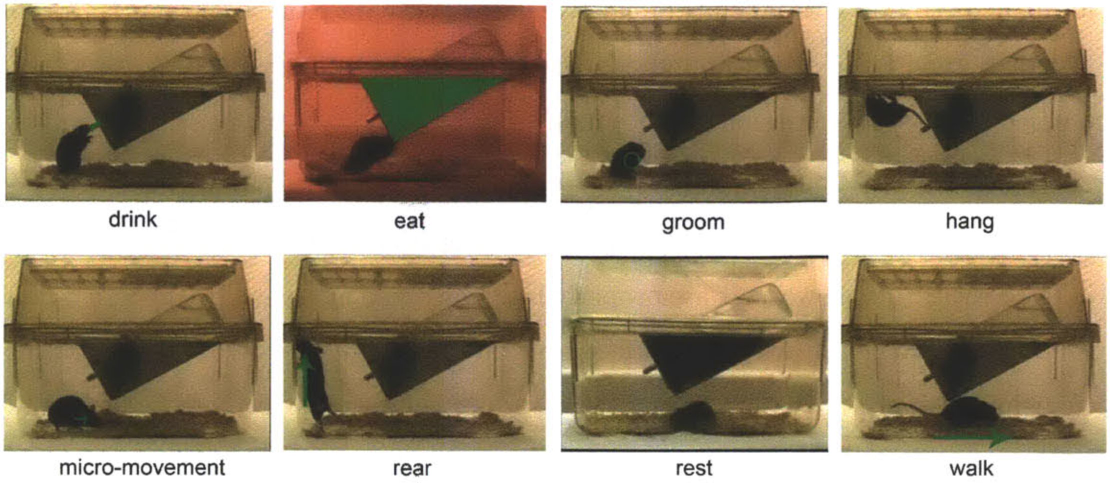


Figure 1-1: Snapshots for the eight home-cage behaviors of interest

1.2.1 Mice behavior recognition task

We want to annotate the following 8 common behaviors of inbred mice (as shown in Figure 1-1):

drink: a mice attaches its mouth on the tip of the drinking tube.

eat: a mice reaches and acquires food from the foodhopper.

groom: a mice has its fore- or hind-limbs sweeping across the face or torso, typically the mice is reared up.

hang: a mice grasps the wire bars with the fore-limbs and/or hind-limbs with at least two limbs off the ground.

rear: a mice has an upright posture and fore-limbs off the ground, and stands against a wall cage.

rest: a mice stays inactive or completely still.

walk: ambulation.

micro-movement: small movements of a mice's head or limbs.

A typical mice behavior recognition task requires an annotator (human or machine) to go through entire video sequence and label all the behaviors of the above 8 types.

1.2.2 Datasets

We use the mice video datasets provided by H. Jhuang and T. Poggio [29, 31, 30, 58]. They collect two datasets: a clipped dataset and a full dataset. The clipped dataset contains 4200 clips with the most exemplary instances of each behavior from 12 videos. These videos contains different mice (differ in coat color, size, gender, etc.) recorded at different times during the day and night during 12 separate sessions. Each clip contains one single behavior. The full dataset contains 12 distinct videos of 30 minutes to 1 hour in length. Every frame of the videos are labeled by two separate groups of people, which results in a total of over 10 hours of continuously annotated videos.

1.3 Online video labeling system

As mentioned in the beginning of this chapter, one question we would like to answer is how to build an efficient online behaviorial annotation tool. Design an online video labeling tools is not easy. Previous attempt includes LabelMe Video from Jenny Yuen et al [72] and VATIC from Carl Vondrick[15]. Most of the tools share the same interface that basically asks annotators to view through a long video and identify the behavior segments. Annotators need to specify both the boundary and the name of each labels. To study the efficiency of the these conventional video labeling method, we develop our own web labeling tool based on Adobe Flash. After performing extensive experiments with web labeling tools, we find that the tool provide a horrible labeling experience, and therefore not a suitable for low-paid online annotators. In our opinion, the key to improve the user experience of the video labeling is to reduce the amount of actions each labeler perform and make the objective clear. As we can learning from the the other mechanical tasks such as image tagging and spam filtering, the high quality tasks are usually simple and concise. Therefore, we consider the second approach by breaking the videos into pieces of tiny segments either uniformly or by using some video segmentation algorithms, and then ask online workers to annotate each pieces by simply assigning an appropriate label. This significantly simplifies the annotation processes and shorten the time for obtaining the label. The design details of our online labeling tools is covered in the Chapter 2.

1.4 Rating the annotators and aggregating the behavioral labels

Once we have results from online labeling tools, we need to build a classifier to predict the correct label for each video based on the multiple labeler's responses. This is the second question we asked at the beginning, i.e. how can we infer the groundtruth if the results we get from online workers are noisy? In the multi-labeler problems, predicting the label purely base on simple majority vote, without regard for the label source properties may not be effective in general. The reasons for the include: some annotators may be more reliable than others, some may be malicious, some may be corrected with others, there may exist different prior knowledge about annotators. Probabilistic methods provide a principled way to approach the problems using standard inference tools. We explore one such approach by formulating a probabilistic graphical model of the labeling process. This will be covered in Chapter 3.

1.5 Related Work

1.5.1 Systems for mice behavior analysis

The previous work on the automatic mice behavior analysis falls into two groups: sensor-based approaches and video-based approaches.

Popular sensor-based approaches include the use of PVDF sensors [6], infrared sensors [26, 25, 47, 59], RFID transponders [43], and photobeam [23]. These approaches have been successfully applied to the analysis of coarse locomotion activity as a proxy to measure global behavioral states such as active v.s. rest. Nevertheless, the physical measurements obtained from these sensor-based approaches are usually not precise and hence limit the complexity of the behavior that can be measured. This problem remains even for commercial systems using transponder technologies such as the intelliCage system (NewBehavior Inc.). While such systems can be effectively used to monitor the locomotion activity of an animal as well as other pre-programmed activities via operant conditioning units located in the corners of the cage, such systems alone cannot be used to study natural behaviors such as grooming, sniffing, rearing or hanging, etc.

The other solution to address the problems described above is to rely on vision-based techniques. Several computer vision systems for tracking mice have been developed[39, 12, 4, 50, 14, 61, 35]. As for sensor-based approaches, these systems are not suitable for the analysis of fine animal activities such as grooming or rearing. The first effort to build an automated computer vision system for the recognition of mouse behaviors is initiated at University of Southern California. As part of its SmartVivarium project, an initial computer vision system is developed for both tracking mice [12] and recognizing the behaviors(eating, drinking, grooming, exploring and resting) of mice[46]. Xue and Henderson also describe an approach [70, 33] for the analysis of rodent behaviors; however, the system is only tested on synthetic data [33] and a very limited number of behaviors. Recently, H. Jhuang, at el.[29, 31, 30, 58] develop a trainable, general-purpose, automated and potentially high-throughput system for the behavioral analysis of mice in their home-cage, base on a computational model of motion processing in the primate visual cortex. They also provide a very large database of manually annotated video sequences of single-mouse behaviors. Besides, X. Burgos-Artizzu, at el. [69] propose a method for analyzing social behavior, which segments continuous videos into action "bouts" by building a temporal context model that combines features from spatio-temporal energy and agent trajectories. Still, these automatic algorithms still perform poorer than human annotators do. This motivate us to search for alternative human-based solutions.

1.5.2 Existing video annotation tools

With the rising popularity and success of massive data sets in vision, the community has put great effort into designing efficient visual annotation tools. Deng et al [20] introduce a crowdsourced image annotation pipeline through ImageNet. Torralba et al [54] present LabelMe as an open platform for dense polygon labeling on static images. Everingham et al [24] describe a high quality image collection strategy for the PASCAL VOC challenge. Von Ahn [66] and Dabbish and Von Ahn [65] et al discover that games with a purpose could be used to label images. Ni et al [45] also propose to combine image labeling with a popular puzzle game. Ramanan [53] et al show that exploiting temporal dependence in video can automatically build a data set of static faces. Welinder et al [48] propose a quality control mechanism for annotation on crowdsourced marketplaces. Vittayakorn and Hays [64] propose quality control measure without collecting more data. Endres et al [22] investigate some of the challenges and benefits of building image datasets with humans in the loop.

However, the same principles that assist and motivate users to annotate static images do not apply to dynamic videos directly. Consequently, significant work has been completed in order to build specialized interfaces tailored for video annotation. Yuen et al [72] introduce LabelMe video, an online, web-based platform that is able to obtain high-quality video labels with arbitrary polygonal paths using homography preserving linear interpolation, and can generate complex event annotations between interacting objects. Mihalcik and Doermann [44] describe ViPER, a flexible and extensible video annotation system optimized for spatial labeling. Huber [16] designed a simplified interface for video annotation. Ali et al [8] present FlowBoost, a tool that can annotate videos from sparse set of key frame annotations. Agarwala et al [7] propose using a tracker as a more reliable, automatic labeling scheme compared to linear interpolation. Buchanan and Fitzgibbon [13] discuss efficient data structures that enable interactive tracking for video annotation. Fisher [52] discusses the labeling of human activities in videos. Smeaton et al [56] describe TRECVID, a large benchmark video database of annotated television programs. Laptev et al [42] further show that using Hollywood movie scripts can automatically annotate video data sets. More recently, Vondrick et al. [15] release VATIC(Video Annotation Tool from Irvine, California), an open source platform for monetized, high quality, crowdsource video labeling.

We first use the conventional way to build video labeling tool and ask user to annotate through entire video. It turns about that there are a few problems of this method: (1) labeling process is very complicated, since annotators need to provide both the boundary and the name for each label; (2) people tend to disagree with each other on the boundaries. (3) videos often do not get fully annotated, as labelers can easily skip some frames of the video. Therefore, we design a novel way of video behavioral labeling mechanism by pre-breaking a long video down to tiny clips using some video segmentation algorithms and asking the annotators to provide a single label description for each clip. This significantly simplify the labeling process.

1.5.3 Crowdsourcing algorithms

Once we obtain the results, we need to predict the groundtruth of the labels based on the multiple annotators' responses. A naive approach to identify the correct answer from multiple workers' responses is to use majority voting. Majority voting simply chooses what the majority of workers agree on. When there are many spammers, majority voting is error-prone since it counts all the workers equally. In general, efficient aggregation methods should take into account the differences in the workers' labeling abilities.

A principled way to address this problem is to build generative probabilistic models for the annotation processes, and assign labels using standard inference tools. To infer the answers of the tasks and also the reliability of workers, Dawid and Skene [19] proposed an algorithm based on expectation maximization (EM) [5]. This approach has also been applied in classification problems where the training data is annotated by low-cost noisy "labelers" [36, 62].Recently, significant efforts have been made to improve performance by incorporating more complicated generative models. For example, Whitehill et al. [34] propose a probabilistic model for image classifications and use it to simultaneously infer the label of each image, the expertise of each labeler, and the difficulty of each image. Welinder and Perona [49] propose a model of the labeling process which includes label uncertainty, as well as multi-dimensional measure of the annotators' ability and derive an online algorithm that estimates the most likely value of the labels and the annotator abilities. It finds and prioritizes experts when requesting labels, and actively excludes unreliable annotators. Based on labels already obtained, it dynamically chooses which images will be labeled next, and how many labels to request in order to achieve a desired level of confidence. Later in [48], they extends to work by introducing a more comprehensive and accurate model of the human annotation process and provide insight into the human annotation process by learning a richer representation that distinguishes amongst the different sources of annotator error. Also, Yan et al. [71] introduce a novel dependency that the annotators's expertise varies depending on the data they observe. That is, an annotator may not be consistently accurate across the task domain.

However, EM is widely criticized for having local optimality issues [21]. In particular, algorithms require an initial starting point which is typically randomly guessed. The algorithm is highly sensitive to this initialization, making it difficult to predict the quality of the resulting estimate; this raises a potential tradeoff between more dedicated exploitation of the simpler models, either by introducing new inference tools or fixing local optimality issues in EM, and the exploration of larger model space, usually with increased computational cost and possibly the risk of over-fitting.

On the other hand, variational approaches [51], including the popular belief propagation (BP) and mean field (MF) methods, provide powerful inference tools for probabilistic graphical models[67, 37]. These algorithms are efficient, and often have good local optimality properties or even globally optimal guarantees [68].

1.6 Contribution

The main contributions of this thesis are the followings:

1. Developed a novel clip-based video labeling tool, which greatly simplifies the traditional video labeling task without comprising much the labeling accuracy. We believe that our video labeling tool is more suitable for crowdsourcing video annotation task, which requires simplicity and clearness.

2. Proposed probabilistic inference methods for label aggregation that simultane-

ously predicts the expertise of the workers and groundtruth of labels. We show that our methods outperform majority vote heuristic in most cases.

3. Designed and implemented a complete system to crowdsource the behavior label for mice videos.

Chapter 2

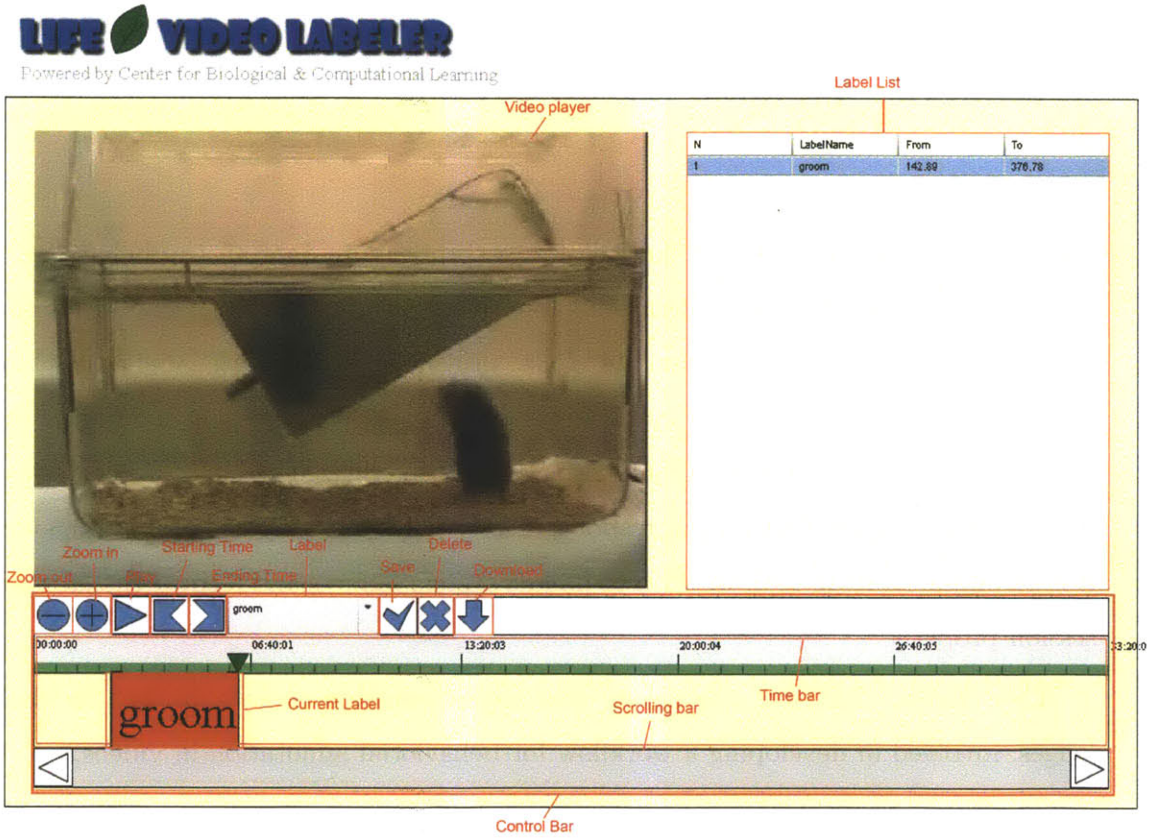
Online Video Labeling Tools

We aim to design an interface that allows workers to annotate all the behaviors of interest in a video. The users should be able to specify the starting frame(time) and the ending frame(time) of a behavior and also provide an appropriate label (name) for that behavior. They also need to be able to perform operations such as modifying the name and the time boundaries of an existing label, and deleting a label. Some desired features of the tool include speed, responsiveness, and intuitiveness. In addition, since it is an online labeling tool, we wish to handle system failures and recover the labeling session properly.

This chapter describes the design and implementation choices, as well as challenges, involved in developing a workflow for behaviorial annotation in videos.

2.1 First design: conventional video labeling tool

In this section, we present our initial design of video labeling system as shown in Figure 2-1. The labeling tool is developed using Adobe Flash CS5. The design follows the philosophy of many existing video labeling tools such as LabelMe Video [72] and VATIC [15]. We believe that the tool satisfies the requirements for crowdsouring behaviorial video labels. In the next two sections, we describe several aspects of our system including the user interface, backend system, and potential disadvantages.



N	LabelName	From	To
1	groom	142.89	376.78

Figure 2-1: Flash video labeling tool interface

2.1.1 Easy-to-use user interface

The system has a very clear user interface that comprises three main area: Video Player (Top-left), Label List (Top-right), and Control Bar (Bottom). the Video Player displays the video that is currently annotated. the Label List displays all the labels that have been previously saved. The Control Bar offers most of the functions for user to annotate the video.

To play and pause a video, click the "Play/Pause" button.

To insert a label, a user need to 1. specify the starting point and ending point of the label by clicking the "Starting time" and "Ending time" buttons respectively, 2. select a label name in the "Label" menu, 3. and click on "Save". During the insertion process, the label is displayed as rectangular box in the time bar. For example, in Figure 2-1 the label "groom" is the current label to be inserted. Once a label is inserted, it is added to the "Label List".

To delete a label, a user can first select a label in the "Label List", which results the label appearing in the "Time bar" as a rectangular box. Then the user can click "Delete" button to remove the label.

To update a label, similar to deletion, a user need to specify the label to be removed by selecting it in the "Label List". And then the user can re-define the starting time, ending time and label name using appropriate buttons. At the end, click "Save" to update the label.

In addition, the labeling tool let users easily find the precise time point using the "Zoom in" and "Zoom out" buttons. Users can also download the labels easily via the "Download" button.

2.1.2 Robust back-end system

We use Apache server [2] and MySQL [3] to support our labeling tool and store all the data including user information, annotations, videos, dataset information, etc. The system allows multiple online workers to annotate the same video and save their responses separately. Therefore, we need to store the responses together with the

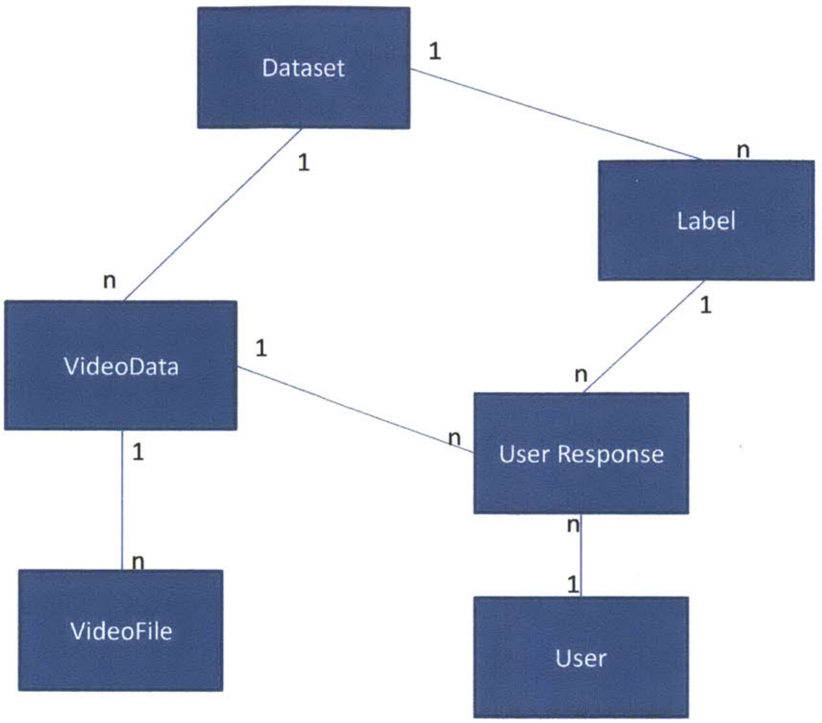


Figure 2-2: Object relation graph of the database model

corresponding user information. In other word, we need to store the relations between users and responses, etc. To satisfy the requirements, we design database tables according to the object relation graph as shown in Figure 2-2. Each box represents a database table and the edges represent the relationships between the tables. The "1" and "n" on the two ends of a edge indicates a foreign key relationship or a "1:n" relationship. For example, the "1:n" relationship between User and User Response means that a user can give multiple responses, while a response can only associate to a single user.

2.1.3 Potential disadvantages

There are some disadvantages of this labeling tool. First, users are asked to specify the boundary of the labels themselves, which makes labeling process very show. Besides, as indicated in the user study, users are not accurate at estimating the best boundary of the label. In fact, they tend to disagree with each other on the boundaries. Moreover, since the labeling process is relatively unconstrained, i.e. users can define a label anywhere in the video, users can easily skip frames and hence give incomplete annotations. Finally, the entire annotation process is still to complex for Mechanical Turk workers as they need to go through a few steps to save a label. Aware of the above problems, we re-think the design of the labeling tool and develop the clip-based video labeling system which will be discussed in the next section.

2.2 Second design: clip-based video labeling

The motivation of designing the a clip-based video labeling system is to simplify the labeling process of conventional labelers. Instead of asking the users to specify the boundaries of the labels, we fix the boundaries for them so that the users only need to provide a name for the label. We denote each of these predefined video segments as a **behaviorial clip** . This significantly simply the labeling process, as most labelers have trouble deciding the two ends of a label. In addition, we merely ask labelers to watch a sequence of behaviorial clips rather than going through a whole video. We believe that this way help workers easily focus on the each video clips and identify the corresponding behaviors. Moreover, since the users are required to go through each clip (pre-defined label) and provide a label name, they are unlikely to skip frames.

The clip-based video labeling system consists of three parts: breaking long videos into behaviorial clips, crowdsourcing to assign annotations to all behaviorial clips, and aggregating and assigning labels to the original videos. To divide a video into small segments, the simplest way is to cut it uniformly under the assumption that if we cut the video into pieces small enough, each piece contains only a single behavior. Besides, we can also use more advanced motion segmentation algorithm to divide the videos. The segmentation methods are discussed in details in **Chapter 4**. We design a labeling tool that allows Mechanical Turk workers to label each behaviorial clip easily, which is explained thoroughly in the remaining part of this section. Last but not the least, we need to develop machine learning algorithms to combine the responses we get from the online workers to infer the most appropriate labels for the original videos. This will be covered in Chapter 3 and Chapter 4.

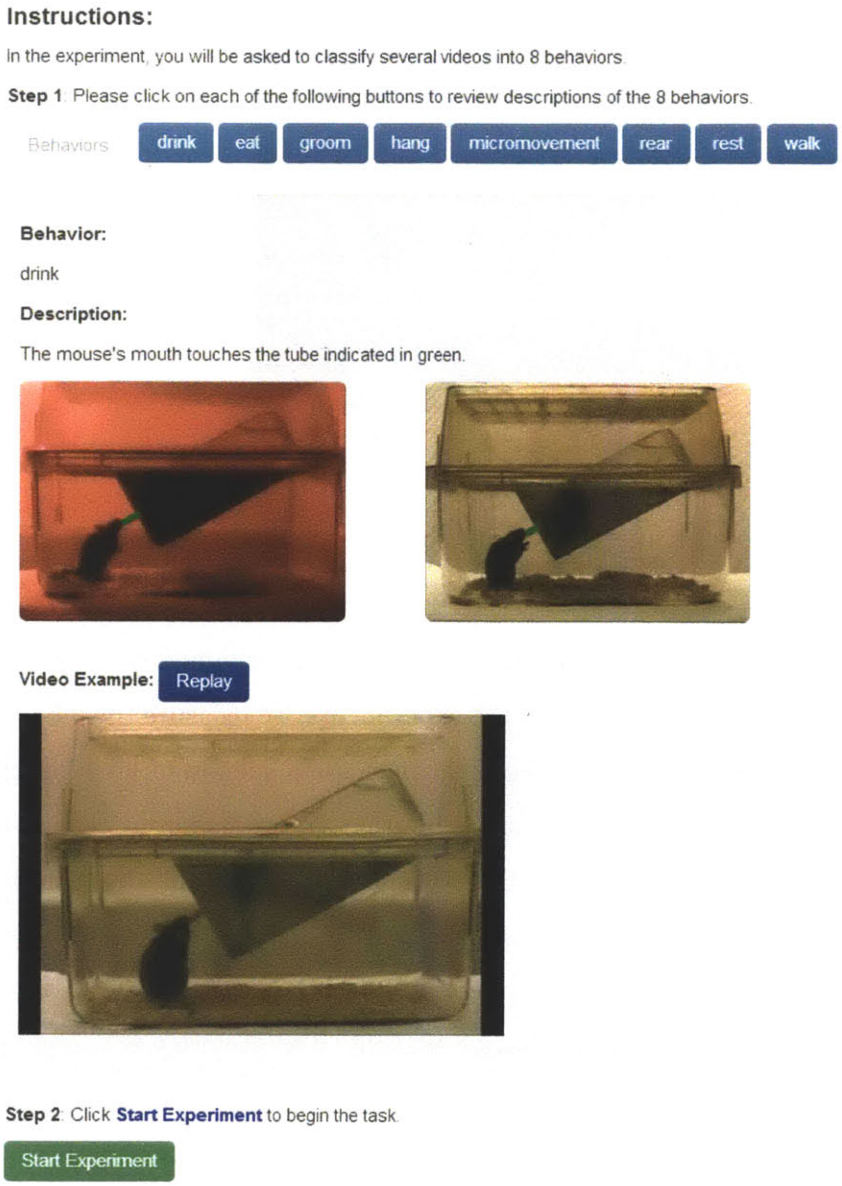
2.2.1 User interface

The user interface of our clip-based video labeling tool consists of two separate parts: instruction page and labeling page as shown in Figure 2-3 and Figure 2-4 respectively. The instruction page contains two simple and clear steps. In Step 1, users are given detailed description and video example of each behavior. They can easily preview any behavior by clicking the corresponding button at the top. Once the online workers are familiar with all the behavior, they can proceed to step 2 which will start the actual labeling process (by clicking the "Start Experiment" button).

The labeling page (Figure 2-4) only includes a video player and label selection bar. To annotate a video clip, the online worker just need to choose a behavior and click "Confirm". Besides, the labeling tool also provides some additional features such as a "Replay" button and video number indicator. The "Replay" button allows users to repeat the video; and the video number indicator tells them the progress of the labeling task. We believe that this simple design will encourage more online workers to complete our tasks.

2.3 User study

To evaluate the usability of the two labeling tools, we perform extensive user study. In this section, we presents some study results. We locate research subjects by hiring dedicated users in MIT Brain and Cognitive Science department. Our dedicated users are experts in computer vision, neuroscience and biology. We also conduct experiments on Amazon Mechanical Turk evaluate the performance of the clip-based labeling tool. We compensate the online workers for $0.12 per 30 video clips. In both cases, we ask the users to label the same 10min mice video. To evaluate clip-based labeling tool, we first break the video into 180 behaviorial clips of 3 seconds length.



Instructions:

In the experiment, you will be asked to classify several videos into 8 behaviors.

Step 1: Please click on each of the following buttons to review descriptions of the 8 behaviors.

Behavior:

drink

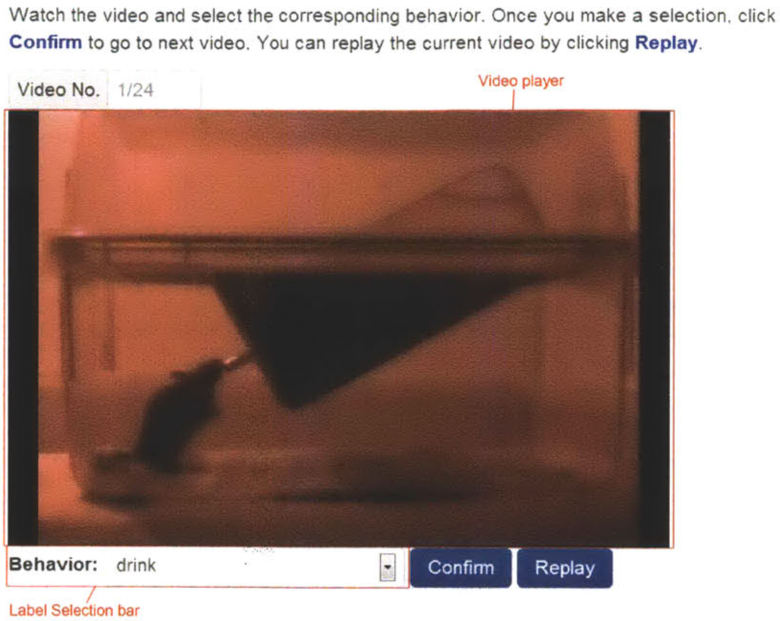
Description:

The mouse's mouth touches the tube indicated in green.

Video Example:

Step 2: Click Start Experiment to begin the task.

Figure 2-3: The instruction of clip-based video labeling tool



Watch the video and select the corresponding behavior. Once you make a selection, click **Confirm** to go to next video. You can replay the current video by clicking **Replay**.

Video No. 1/24

Behavior: drink

Figure 2-4: The user interface of clip-based video labeling tool

Then we randomized the order of the clips presented to the users, a necessary step to reduce a learning and memorization bias. In the experiments, we measure the labeling tools from the various aspects including the total time users spends in labeling the entire video, the coverage of the labels provided by users (i.e. how many percent of frames are labeled by users?), the number of labels provided by users, and the accuracy of the users. Note the to measure the accuracy, we used the groundtruth annotations provided in the dataset [29, 31, 30, 58]. The groundtruth labels are provided by a group of expert annotators in the mice behavior analysis. The accuracy is computed by comparing the users' label with the groundtruth on every frame of the video (The frames that are not annotated by users are considered to be labeled incorrectly).

The study results for our conventional labeling tool are shown in the Table 2.1. The tables shows the total time, label coverage, number of labels, and accuracy of six labelers. Note that it takes in average 2045.6 seconds (more than half an hour) to label a 10-minute video. And the workers only label 91.25% of the frames in average and have average accuracy 62.99%. Among all labelers, Labeler D has the worst

Subject	Total Time (s)	Label Coverage(%)	No. of Labels	Accuracy(%)
A	2146.3	97.4	100	63.8
B	1911.1	90.6	78	65.0
C	2141.9	94.1	125	69.1
D	1848.0	75.2	82	47.0
E	2133.1	94.7	99	63.1
F	2093.4	95.6	127	69.9
Mean	2045.6 (±131.5)	91.3	102	63.0 (±8.3)

Table 2.1: The user study result of the conventional labeling tool

performance (47.0% accuracy). We can see that he has difficulty to finding behaviors in the video and only annotate those most distinguishable parts, as his labels only cover 75% of the total frames. Besides, we also plot the agreement matrix of the labelers, which measures the ratios of the frame labels the labelers agree with each other(as shown in Figure 2-5). We see that even labelers with similar accuracy do not agree with each other on about 20%-30% of the frame labels.

Now we look at the study results for clip-based labeling tool, which are shown in the Table 2.2. One result we notice immediately is that total time required to label the 10min video drops from 2045.6 seconds to 1117.5 seconds in average. With the clip-based labeling tools, it save the workers about half of the time to label the same amount of the videos. Since the workers are required to go through all the clips in the clip-based labeling, the label coverage goes up to 100%. Besides, the average accuracy of the labelers increases by 1.2% as compared to that using conventional labeling tools. Notably, Labeler D achieves about 12% accuracy increase. Moreover, if we plot the agreement matrix of the labelers (as shown in Figure 2-5) and compare it to the agreement matrix in Figure 2-5, we can see clearly that the overall agreement among labelers increases. To put it quantitatively, the average agreement increases by 11.6% (The average agreement of the labelers is computed by the summation of the agreement of all pairs divided by the total number of pairs.). Therefore, using clip-based labeling tool, we can obtain more consistent results from the labelers.

We also ask Mechanical Turk workers to label the same video using clip-based labeling tool. Each behaviorial clip is annotated by 10 people. We simply use majority vote to decide the best label for each clip (Better algorithms will be discussed in the

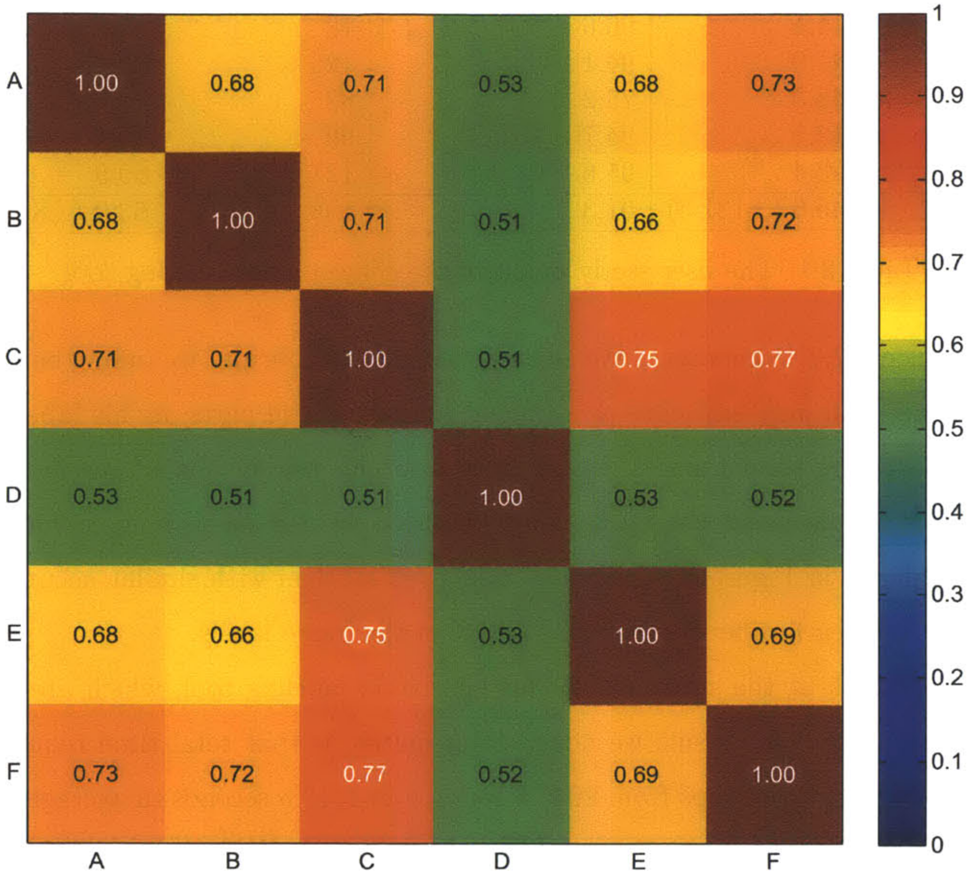


Figure 2-5: Agreement Matrix of the conventional labeling tool: The entry (i, j) indicates the percentage of frames Annotator i agrees with Annotator j.

Subject	Total Time (s)	Label Coverage(%)	No. of Labels	Accuracy(%)
A	1120.0	100	180	67.0
B	891.0	100	180	68.0
C	885.0	100	180	65.4
D	1358.0	100	180	59.0
E	1337.0	100	180	60.0
F	1114.0	100	180	65.0
Mean	1117.5 (±205.6)	100	180	64.2 (±3.8)

Table 2.2: The user study result of the clip-based labeling tool

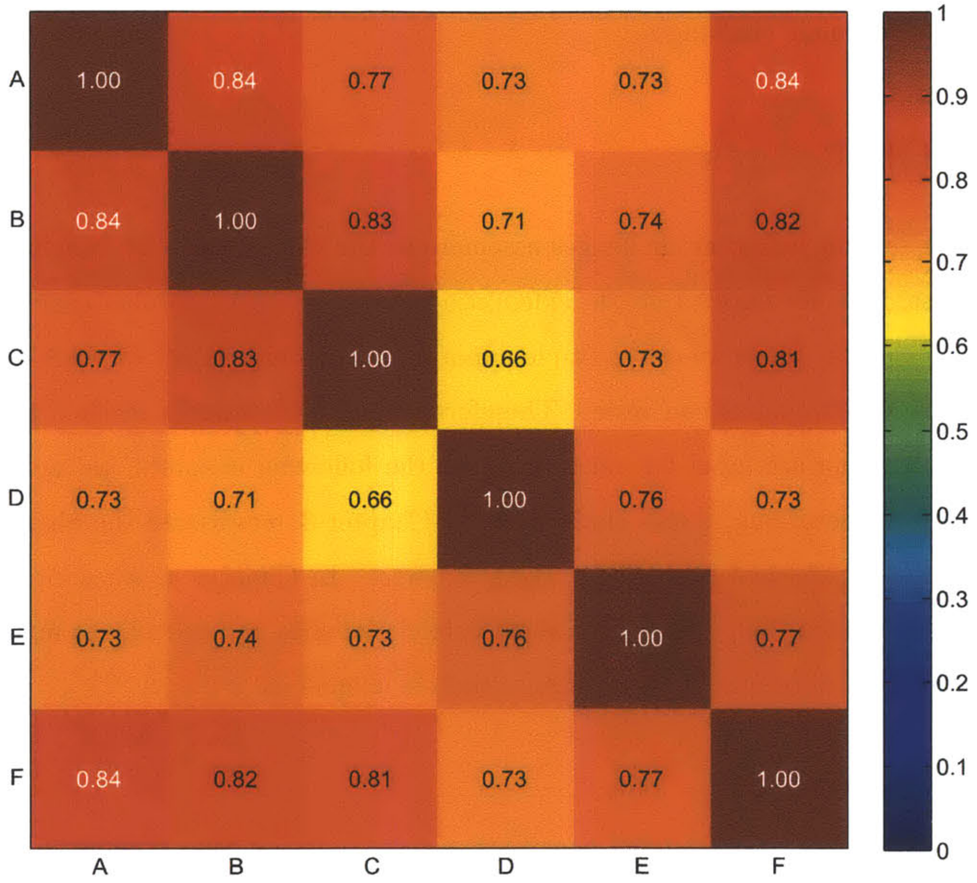


Figure 2-6: Agreement Matrix of the clip-based labeling tool: The entry (i, j) indicates the percentage of frames Annotator i agrees with Annotator j.

later chapters.). We obtain the labeling accuracy of 62.63%, and the average labeling time 1433 seconds, which is computed by the total summation of time spent on all the clips over 10 the number of annotations per clip.

In summary, the user study shows that clip-based labeling tool improve the conventional labeling tool on the label coverage, total labeling time, while achieving the similar labeling accuracy. However, the clip-based labeling tool also presents us some interesting technical challenges.

2.4 Challenges

There are several technical challenges associate to the clip-based video labeling system. First, it is not easy to divide video into pieces that each contains exactly one behavior. Besides, since we ask multiple annotators to provide labels for each behaviorial clips, the responses can varies. Therefore, we need to design a method to infer the most appropriate label for each clips. In the following chapters, we are going to explore the solutions to this challenges. In Chapter 3, we discuss the algorithms to infer correct answers from the workers' answers. In Chapter 4, we address the methods to segment long videos into short behaviorial clips and the way to make use of the temporal information to aggregate workers' responses.

Chapter 3

Video Annotation for independent clips

3.1 Modeling Annotators and Labels

Consider a set of N videos clips denoted by $\mathbf{I} = \{1, \ldots, N\}$. Each video belongs to one of D possible categories (e.g. the eight behavioral labels in our experiment.). We wish to determine the groundtruth class label $\mathbf{z}_i \in \{1, \ldots, D\}$ of each video i. We use $\mathbf{z}$ to denote the set of all the groundtruth labels $\{\mathbf{z}_i\}_{i \in I}$. The observed labels depend on several causal factors: (1) the expertise of labeler and (2) the true label. We model the expertise of the annotator j by a vector of parameter $\mathbf{a}_j$. For example, it can be scalar, $\mathbf{a}_j = a_j$, where $\mathbf{a}_j \in [0, 1]$. Here an $\mathbf{a}_j = 1$ means the labeler always labels images correctly; $\mathbf{a}_j = 0$ means the labeler always labels the images incorrectly. There are M annotators in total, denoted by $\mathbf{A} = \{1, \ldots, M\}$, and the set of their parameter vectors is $\mathbf{a} = \{\mathbf{a}_j\}_{j=1}^{M}$. Each annotator j provides labels $\mathbf{Y}^\mathbf{j} = \{\mathbf{y}_{ij}\}_{i \in \mathbf{I}_j}$ for all or a subset of the videos, $\mathbf{I}_j \subseteq \mathbf{I}$. Likewise, each video i has labels $\mathbf{Y}_i = \{\mathbf{y}_{ij}\}_{j \in \mathbf{A}_i}$ provided by a subset of the annotators $\mathbf{A}_i \subseteq \mathbf{A}$. The set of all labels is denoted $\mathbf{Y}$. For the purpose of our task, we assume the labels $\mathbf{y}_{ij}$ belong to the same set as the underlying groundtruth values $\mathbf{z}_i$.

Figure 3-1 shows a causal model of the labeling process. The observed label $\mathbf{y}_{ij}$ depends on true video labels $\mathbf{z}_i$ and the labeler accuracy values $\mathbf{a}_j$. And $\mathbf{z}_i$ and $\mathbf{a}_j$ are

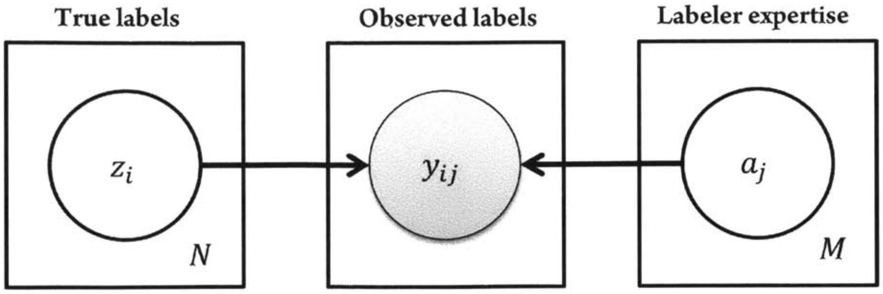


Figure 3-1: Graphical model of true labels, observed labels, and labeler expertise

independent. If we do not assume any prior on $\mathbf{a}_j$, the join probability distribution can thus be factorized as

$$p(\mathbf{Y}, \mathbf{z}, \mathbf{a}) = \prod_{i=1}^{N} p(\mathbf{z}_i) \prod_{\mathbf{y}_{ij}} p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j) \tag{3.1}$$

We can simply assume that the labels $\mathbf{y}_{ij}$ are generated as follows:

$$p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j) = \begin{cases} a_j & \mathbf{y_{ij}} = \mathbf{z_i} \\ \frac{1-a_j}{D-1} & \mathbf{y_{ij}} \neq \mathbf{z_i} \end{cases} \tag{3.2}$$

Thus, the annotator is assumed to provide the correct value with probability a_j and an incorrect value with probability $(1-a_j)$. Here we assume the probability of getting the each of the incorrect labels is the same.

3.2 Expectation Maximization Approach

The observed labels are samples from the $\mathbf{Y}$ random variables. The unobserved variables are the true video labels $\mathbf{z}$, the different labeler accuracies $\mathbf{a}$. Our goal is to efficiently search for the most probable values of the unobserved variables $\mathbf{z}$ and $\mathbf{a}$ given the observed data. To achieve that, we can use Expectation-Maximization approach (EM) to obtain maximum likelihood estimates of the parameters of interest.

E step: Assume we have a current estimate if **a** of the annotator parameters, we need compute the posteriors of all $\mathbf{z}_j$ given the **a** and **Y**:

$$\begin{aligned} p(\mathbf{z}_i|\mathbf{Y},\mathbf{a}) &= p(\mathbf{z}_i|\mathbf{Y}_i,\mathbf{a}) \\ &\propto p(\mathbf{z}_i|\mathbf{a})p(\mathbf{Y}_i|\mathbf{z}_i,\mathbf{a}) \\ &\propto p(\mathbf{z}_i)p(\mathbf{Y}_i|\mathbf{z}_i,\mathbf{a}) \\ &\propto p(\mathbf{z}_i)\prod_{j\in\mathbf{A}_i} p(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j) \end{aligned}$$

where we noted that $p(\mathbf{z}_i|\mathbf{a}) = p(\mathbf{z}_i)$ using the independent assumptions from the graphical model.

M step: To estimate the annotator parameters a, we maximize the expectation of the logarithm of the posterior on a with respect to $p(\mathbf{z}_i)$ from the E-step. We maximize the auxiliary function $Q(\mathbf{a})$ to update **a** as follows:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} Q(\mathbf{a}) \tag{3.3}$$

where

$$\begin{aligned} Q(\mathbf{a}) &= \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{Y}|\mathbf{z},\mathbf{a})] \\ &= \sum_{j=1}^{M} Q_j(\mathbf{a}_j) \end{aligned} \tag{3.4}$$

where $\mathbb{E}_{\mathbf{z}}[\cdot]$ is the expectation with respect to $p(\mathbf{z})$ and $Q_j(\mathbf{a}_j)$ is defined as follows:

$$\begin{aligned} Q_j(\mathbf{a}_j) &= \sum_{i\in\mathbf{I}_j} \mathbb{E}_{z_i}[\log p(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j)] \\ &= \sum_{i\in\mathbf{I}_j}\sum_{\mathbf{z}_i} p(\mathbf{z}_i)\log p(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j) \end{aligned} \tag{3.5}$$

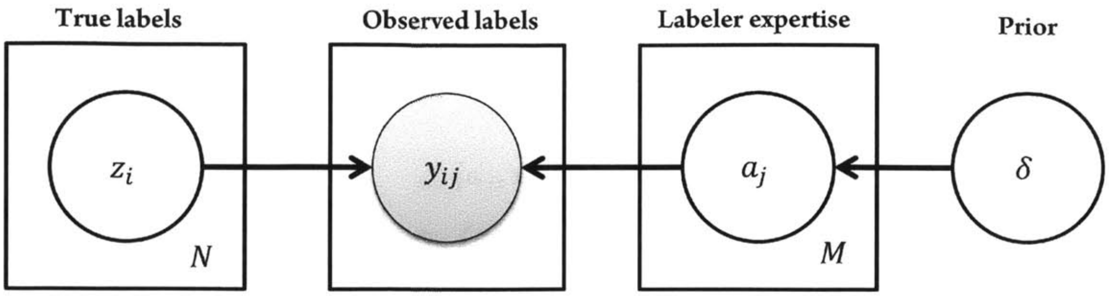


Figure 3-2: Graphical model of true labels, observed labels, and labeler expertise with a prior on each labeler expertise

As we can see from the above, the optimization can be carried out separately for each annotator. We can differentiate Q_j to arrive at:

$$\frac{dQ_j}{d\mathbf{a}_j} = \sum_{i\in\mathbf{I}_j}\sum_{\mathbf{z}_i} p(\mathbf{z}_i)\frac{1}{p(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j)}\frac{dp(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j)}{d\mathbf{a}_j} \tag{3.6}$$

$$= \sum_{i\in\mathbf{I}_j}\left[p(\mathbf{z}_i=\mathbf{y}_{ij})\frac{1}{\mathbf{a}_j} - \frac{1}{D-1}\sum_{\mathbf{z}_i\neq\mathbf{y}_{ij}} p(\mathbf{z}_i)\frac{D-1}{1-\mathbf{a}_j}\right] \tag{3.7}$$

From 3.2 and 3.5, we can see that $Q_j((a)_j)$ is concave. Therefore, we can let $\frac{dQ_j}{d\mathbf{a}_j}=0$, we have a closed-form solution

$$\mathbf{a}_j = \frac{\sum_{i\in\mathbf{I}_j} p(\mathbf{z}_i=\mathbf{y}_{ij})}{|I_j|} \tag{3.8}$$

3.3 Prior on a

We can also assume a prior for each $\mathbf{a}_j$ as in Figure 3-2. Since $\mathbf{a}_j \in [0,1]$, we can use a beta distribution as a prior for $\mathbf{a}_j$. The joint probability distribution becomes:

$$p(\mathbf{Y},\mathbf{z},\mathbf{a}) = \prod_{i=1}^{N} p(\mathbf{z}_i)\prod_{y_{ij}\in Y} p(\mathbf{y}_{ij}|\mathbf{z}_i,\mathbf{a}_j)\prod_{j=1}^{M} p(\mathbf{a}_j|\alpha,\beta) \tag{3.9}$$

Introducing the prior does not change the E-step of the algorithm. However, we need to modify the M-step to take care of the prior.

M step:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} Q(\mathbf{a}) \tag{3.10}$$

where

$$\begin{aligned} Q(\mathbf{a}) &= \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{Y}|\mathbf{z}, \mathbf{a}) + \log p(\mathbf{a}|\alpha, \beta)] && (3.11) \\ &= \sum_{j=1}^{M} Q_j(\mathbf{a}_j) && (3.12) \end{aligned}$$

where

$$\begin{aligned} Q_j(\mathbf{a}_j) &= \log p(\mathbf{a}_j|\alpha, \beta) + \sum_{i \in \mathbf{I}_j} \mathbb{E}_{\mathbf{z}_i}[\log p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j)] \\ &= \log p(\mathbf{a}_j|\alpha, \beta) + \sum_{i \in \mathbf{I}_j} \sum_{\mathbf{z}_i} p(\mathbf{z}_i) \log p(\mathbf{y}_{\mathbf{ij}}|\mathbf{z}_i, \mathbf{a}_j) && (3.13) \end{aligned}$$

Differentiating Q_j, we have

$$\begin{aligned} \frac{dQ_j}{d\mathbf{a}_j} &= \frac{1}{p(\mathbf{a}_j(\alpha, \beta))} \frac{dp(\mathbf{a}_j(\alpha, \beta))}{d\mathbf{a}_j} + \sum_{i \in \mathbf{I}_j} \sum_{\mathbf{z}_i} p(\mathbf{z}_i) \frac{1}{p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j)} \frac{dp(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j)}{d\mathbf{a}_j} \\ &= \frac{(\alpha - 1) - (\alpha + \beta - 2)\mathbf{a}_j}{\mathbf{a}_j(1 - \mathbf{a}_j)} + \sum_{i \in \mathbf{I}_j} \left[p(\mathbf{z}_i = \mathbf{y}_{ij}) \frac{1}{\mathbf{a}_j} - \frac{1}{D-1} \sum_{\mathbf{z}_i \neq \mathbf{y}_{ij}} p(\mathbf{z}_i) \frac{D-1}{1 - \mathbf{a}_j} \right] \end{aligned}$$

By setting the derivative to zero, we obtain

$$\hat{\mathbf{a}}_j = \frac{(\alpha - 1) + \sum_{i \in \mathbf{I}_j} p(\mathbf{z}_i = \mathbf{y}_{ij})}{(\alpha + \beta - 2) + |\mathbf{I}_j|} \tag{3.14}$$

However, the $\hat{\mathbf{a}}_j$ may not maximize the $Q_j(\mathbf{a}_j)$. This is due to the fact that $Q_j(\mathbf{a}_j)$ may not be concave. To find the maximum, we make use of the fact that $Q_j(\mathbf{a}_j)$ has at most one critical point (sometimes $\mathbf{a}_j$ computed by 3.14 falls outside [0, 1]). We can simply compare the Q_j value at $\hat{\mathbf{a}}_j$ with its values at 0 and 1 to obtain the maximum.

3.4 Multidimensional expertise of annotators

In the previous analysis, we use conditional probability

$$p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{a}_j) = \begin{cases} a_j & \mathbf{y_{ij}} = \mathbf{z_i} \\ \frac{1-a_j}{D-1} & \mathbf{y_{ij}} \neq \mathbf{z_i} \end{cases} \tag{3.15}$$

which assumes that a user have the same probability a_j of getting the correct answer and same probability $\frac{1-a_j}{D-1}$ of getting each incorrect answer regardless of the true label of the video. This assumption may not be true in general, since annotators may have different areas of strength, or expertise, and thus provide more reliable labels on different subsets of videos. For example, when ask to differentiate the 8 behaviors in the videos, some annotators may be more aware of the distinction between eat and drink, while other may be more aware of the distinction between groom and micromovement.

To capture the variation of expertise, we may define $\mathbf{a}_j = \mathbf{A}_j$, the confusion matrix. Each entry $\mathbf{A}_j(s,t)$ in a confusion matrix is the probability with which a clip is annotated as t when its true label is s, as computed by

$$\mathbf{A_j}(s,t) = \frac{\text{\# total clips labeled as } s \text{ by annotator } j \text{ where its groundtruth is } t}{\text{\#total clips labeled by annotator } j \text{ where its grouthtruth is } t}.$$

Using the new formulation of $\mathbf{a}_j$, the E step becomes:

$$p(\mathbf{z}_i|\mathbf{Y}, \mathbf{A}) \propto p(\mathbf{z}_i) \prod_{j \in \mathbf{A}_i} p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{A}_j) \tag{3.16}$$

where we use the conditional probability

$$p(\mathbf{y}_{ij}|\mathbf{z}_i, \mathbf{A}_j) = \mathbf{A}_j(\mathbf{y}_{ij}, \mathbf{z}_i). \tag{3.17}$$

In the M-step, we need to find

$$\mathbf{A}_j = \arg\max_{\mathbf{A}_j} Q_j(\mathbf{A}_j) \tag{3.18}$$

for all $\mathbf{A}_j$, where

$$Q_j(\mathbf{A}_j) = \sum_{i \in \mathbf{I}_j} \sum_{\mathbf{z}_i} p(\mathbf{z}_i) \log p(\mathbf{y}_{ij} | \mathbf{z}_i, \mathbf{A}_j) \tag{3.19}$$

Since

$$\mathbf{A}_j = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{pmatrix} \tag{3.20}$$

where $\sum_{s=1}^{n} a_{st} = 1$, $\forall t$. Therefore, we have $a_{nt} = 1 - \sum_{s=1}^{n-1} a_{st}$ for $t = 1, \dots, n$. It means that $\mathbf{A}_j$ has $n \times (n-1)$ dimensions. Taking the derivative of 3.19 in each dimension, we have

$$\begin{aligned} &\sum_{i \in \mathbf{I}_j} \sum_{\mathbf{z}_i} p(\mathbf{z}_i) \frac{1}{p(\mathbf{y}_{ij} | \mathbf{z}_i, \mathbf{A}_j)} \frac{\partial p(\mathbf{y}_{ij} | \mathbf{z}_i, \mathbf{A}_j)}{\partial a_{st}} \\ = &\sum_{\mathbf{y}_{ij}=s, \mathbf{z}_i = t} \frac{p(\mathbf{z}_i)}{\mathbf{A}_j(s,t)} - \sum_{\mathbf{y}_{ij}=n, \mathbf{z}_i = t} \frac{p(\mathbf{z}_i)}{1 - \sum_{k=1}^{n-1} \mathbf{A}_j(k,t)} \end{aligned} \tag{3.21}$$

for all $s = 1, \dots, n-1$ and $t = 1, \dots, n$. Set the derivatives to zeros. Then for each t, we have

$$\sum_{\mathbf{y}_{ij}=s, \mathbf{z}_i = t} \frac{p(\mathbf{z}_i)}{\mathbf{A}_j(s,t)} = \sum_{\mathbf{y}_{ij}=n, \mathbf{z}_i = t} \frac{p(\mathbf{z}_i)}{1 - \sum_{k=1}^{n-1} \mathbf{A}_j(k,t)} \tag{3.22}$$

for all $s = 1, \dots, n-1$. Since L.H.S of 3.22 stay unchanged for all s, we then have

$$\frac{\sum_{\mathbf{y}_{ij}=s, \mathbf{z}_i = t} p(\mathbf{z}_i)}{\mathbf{A}_j(s,t)} = \frac{\sum_{\mathbf{y}_{ij}=v, \mathbf{z}_i = t} p(\mathbf{z}_i)}{\mathbf{A}_j(v,t)} \tag{3.23}$$

for all $s, v = 1, \dots, n-1$, which implies

$$\mathbf{A}_j(s,t) = \frac{\sum_{\mathbf{y}_{ij}=s, \mathbf{z}_i = t} p(\mathbf{z}_i)}{\sum_{\mathbf{z}_i = t} p(\mathbf{z}_i)} \tag{3.24}$$

for all $s = 1, \dots, n-1$, and $t = 1, \dots, n$. And it is also clear that

$$\mathbf{A}_j(n,t) = \frac{\sum_{\mathbf{y}_{ij}=n, \mathbf{z}_i = t} p(\mathbf{z}_i)}{\sum_{\mathbf{z}_i = t} p(\mathbf{z}_i)} \tag{3.25}$$

for all $s = 1, \dots, n-1$, and $t = 1, \dots, n$.

3.5 Initialization

The EM algorithm is a local optimization algorithm, it can only converges to a local optimal. Since the likelihood function of our problem is not a convex function of $\mathbf{a}$ and $\mathbf{z}$, there may exists one or more local maximum points. Thus, the initial guess for our algorithms are very important. In our implementation, we use the result from majority vote algorithm with some perturbations as a starting point for the EM algorithm. This choice of starting point improves the stability of our algorithms.

3.6 Simulation

We explore the performance of the model using a set of video labels generated by the model itself. Since, in this case we know the parameters $\mathbf{z}$, $\mathbf{a}$ that observed labels, we can compare them with corresponding parameters estimated using the EM procedure.

3.6.1 Basic simulation

The first experiment simulate binary annotations, where we simulate between 4 and 20 labelers, each labeling 2000 videos, whose true labels $\mathbf{z}$ are either 1 or 2 with equal probability. The accuracy $\mathbf{a}_j$ of each annotator is generated from a normal distribution with mean 0.6 and variance 0.1, which is chosen under our assumption that adverse labelers (whose accuracy belows 0.5) are rare. Given these labeler abilities, the observed labels $\mathbf{y}_{ij}$ are sampled according to Equation 3.2 using $\mathbf{z}$. Finally, the three algorithms described above, namely basic EM inference procedure, EM algorithm with beta prior (with beta parameters 2 and 2), and EM algorithm with $\mathbf{a}$ being the confusion matrix $\mathbf{A}$, and majority vote algorithm are executed to estimate $\mathbf{a}$, and $\mathbf{z}$. The procedure (including generate synthetic data) is repeated 10 times to smooth out variability between trails.

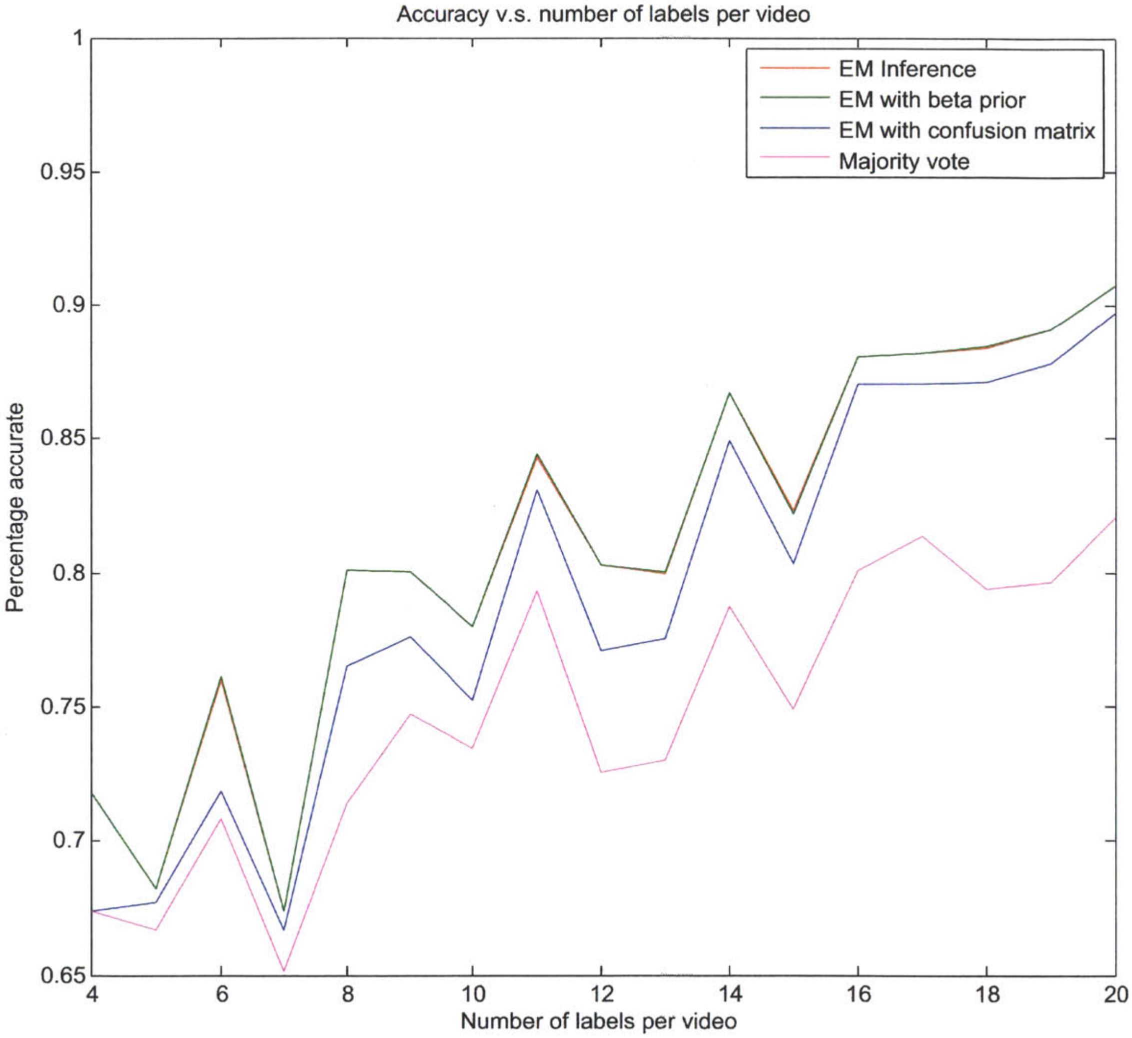


Figure 3-3: Basic simulation result: accuracy of algorithms v.s. number of labels per video

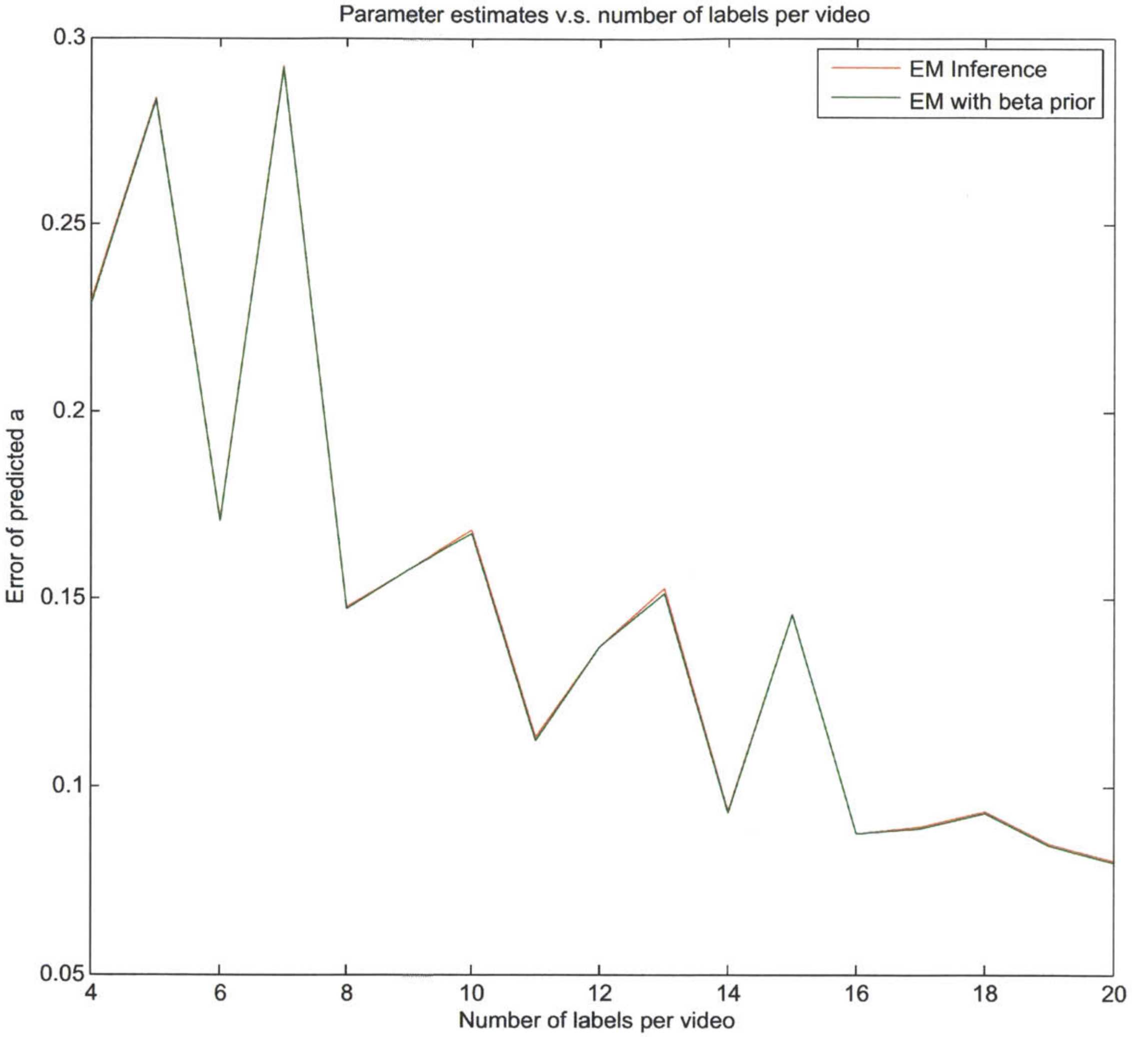


Figure 3-4: Basic simulation result: parameter estimates v.s. number of labels per video

We compute the percentage of predicted labels that matched the true labels. We compare the maximum likelihood estimates of our algorithms to estimates obtained by taking the majority vote as the predicted label. The predictions of the proposed algorithms are computed by taking the most probable label according to the posterior probability $p(\mathbf{z}_i)$ of each video. The results (averaged over all 10 experimental runs) are shown in Figure 3-3. As expected, the overall accuracy increases as number of labelers increase. Our three algorithms achieve higher accuracy than the majority vote heuristic, since our algorithms make use of the fact that some labelers are experts and hence their votes should count more than the votes of less skilled labelers on the same video. Besides, we also compute the predicted labels's accuracies and compare them to the true accuracies. We also evaluate the results from the basic EM approach and the EM with beta prior approach, since only these two algorithm explicitly compute the annotators' overall accuracy. The root mean square errors are shown in Figure 3-4. As expected, as the number of labelers grows, the parameter estimates converge to the true values.

We also run the same experiment again with higher variance of labelers' accuracy, namely 0.4. With higher variance, we allow more adverse labelers. The result is shown in Figure 3-5. As expected, we see more fluctuations in all the curves. One possible explanation is that in the binary annotation as more adverse labelers join the experiment, we have two groups of labelers who stand on opposite sides in most cases. Therefore, both our algorithms and majority vote are confused about which side gives the true label. However, our algorithms still perform better than majority vote in general(sometimes even achieve 15% higher accuracy).

3.6.2 Multi-valued annotations

In this experiment, we would like to evaluate the performance of the algorithms in multi-valued annotations. Similar to the prior experiment, we simulate between 4 and 20 labelers, each labeling 2000 videos, whose true labels $\mathbf{z}$ is selected from the $\{1, \ldots, 8\}$ with equal probability. The accuracy $\mathbf{a}_j$ is drawn from a normal distribution with mean 0.6 and variance 0.1. The observed labels $\mathbf{y}_{ij}$ are sampled according to

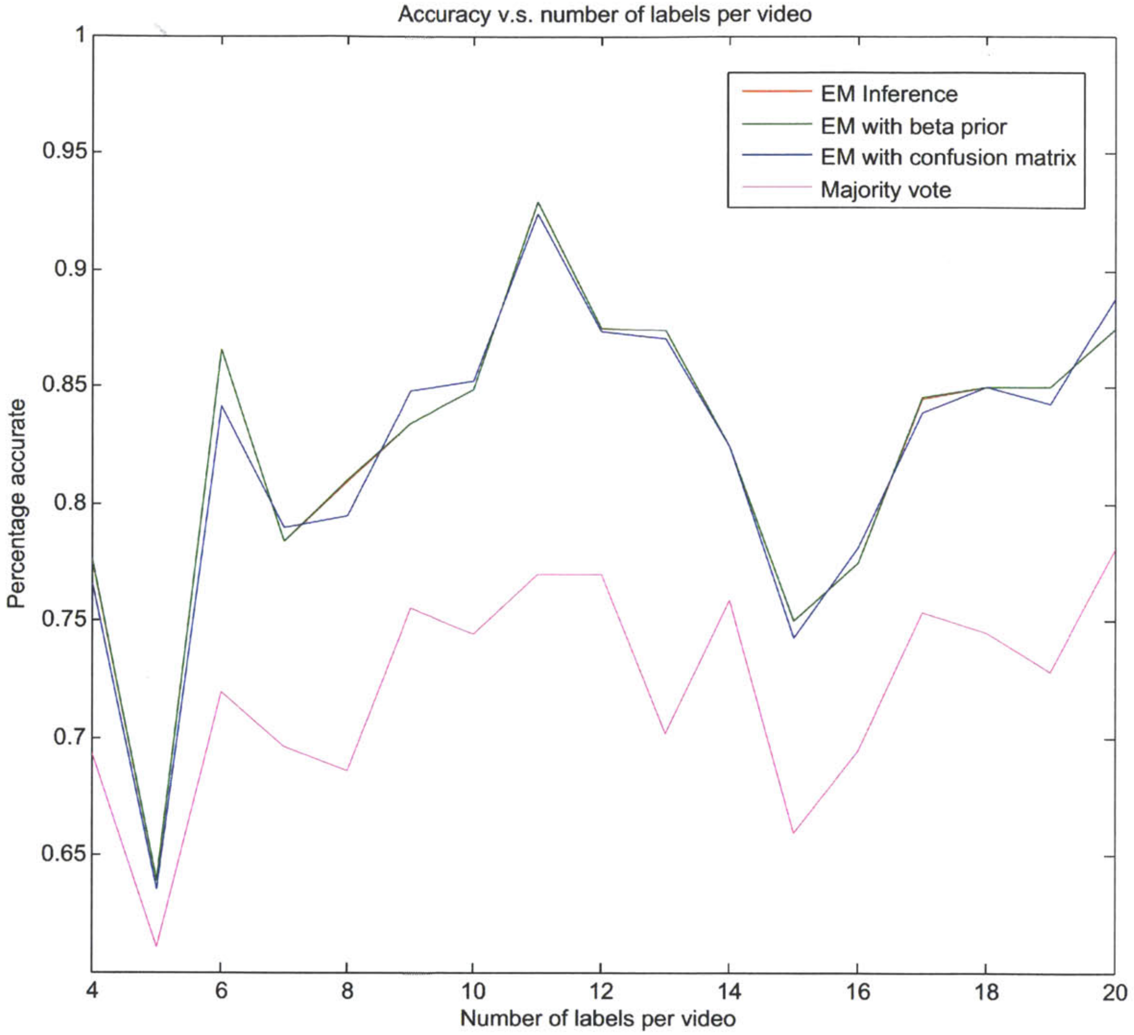


Figure 3-5: Basic simulation result(labelers' accuracy variance 0.4): accuracy of algorithms v.s. number of labels per video

Equation 3.2. The produce is repeated 10 times to produce the average performance.

Figure 3-6 shows the results of the simulation. Our model achieves a consistently lower error rate as compared to majority vote. However, in this case, the improvement is not as significant as that in binary annotation case. We also repeat the same experiment with higher variance of accuracy $\mathbf{a}_j$ (0.4). As can be seen from Figure 3-7, the advantage of our algorithms over majority vote is clearer. The difference is particularly pronounced when the number of labelers per video is small.

3.6.3 Stability of the algorithms under various labelers

For most of the online annotation tasks, we have a fixed dataset and the a group of online workers can varies, since there is impossible to expect a same set of people will always do your task. Thus, in this experiment, we would like to find out the performance of the algorithms subjects to the labelers change in the same dataset. The experiment setting is the same as that in Section 3.6.1, except we only generate the true labels $\mathbf{z}$ once and use it for the 10 repeats. We compute the average labeling accuracy with the standard deviation against different number of labelers. The results are shown in Figure 3-8. To make the graph clear, we only plot the performance of our basic EM inference algorithm (the other two algorithm achieve similar performance) and the majority vote algorithm. As expected, our algorithm outperforms majority vote in terms of average accuracy with similar standard deviation.

3.7 Empirical study: video clip dataset

Now we are ready to experiment with real video label data. We use the video clips dataset from [29, 31, 30, 58] and randomly select 480 video clips. There are 60 video clips for each of the 8 behaviors. The 8 behaviors includes drink, eat, groom, hang, rear, rest, walk, and micro-movement. The details of the task is described in Section 1.2.1. We use the clip-based labeling tool described in Section 2.2 to collect labels from Mechanical Turk.

We obtain labels for 480 videos from 85 different Mechanical Turk labelers. Each

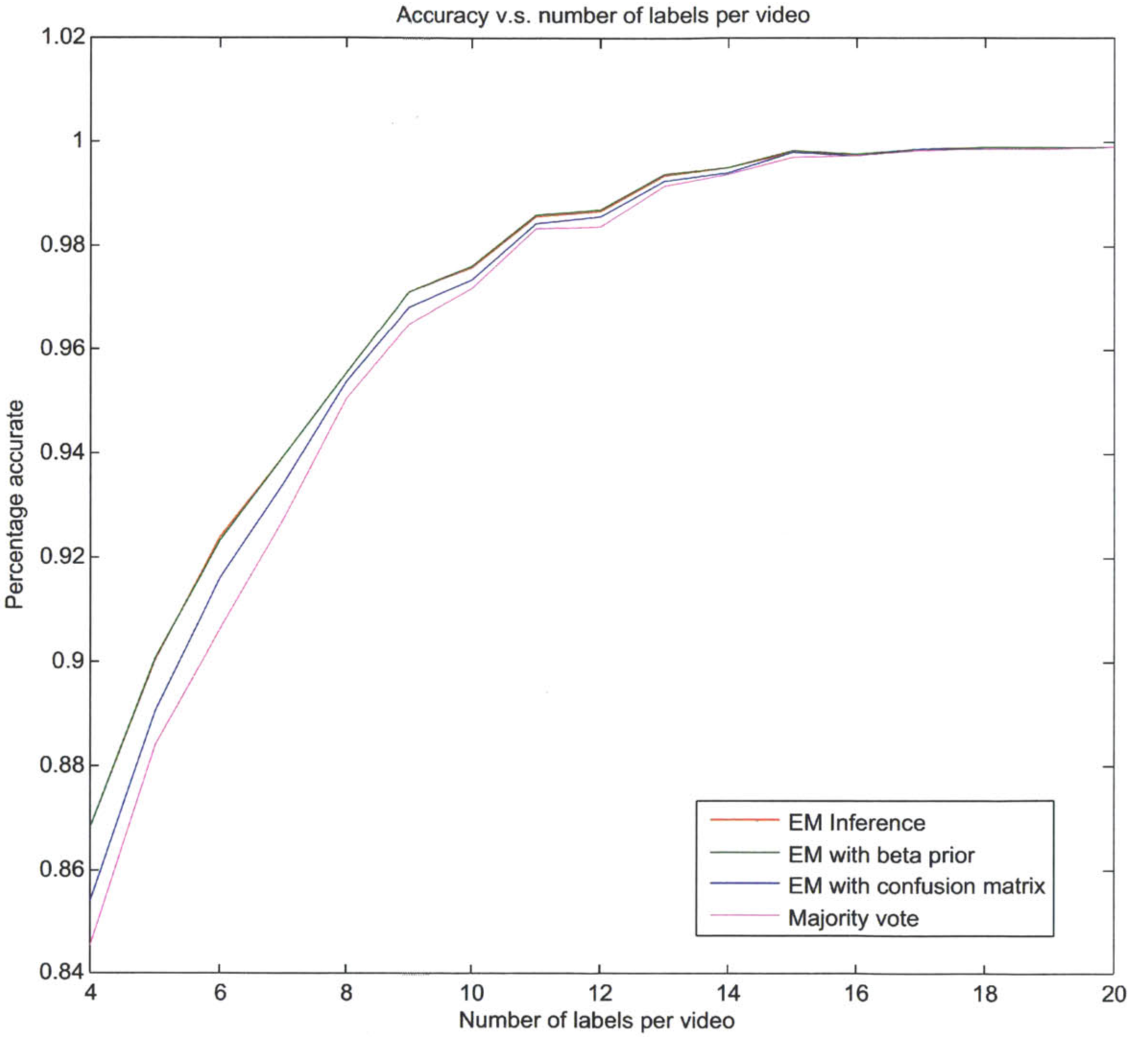


Figure 3-6: Multi-valued annotation simulation result: accuracy of algorithms v.s. number of labels per video

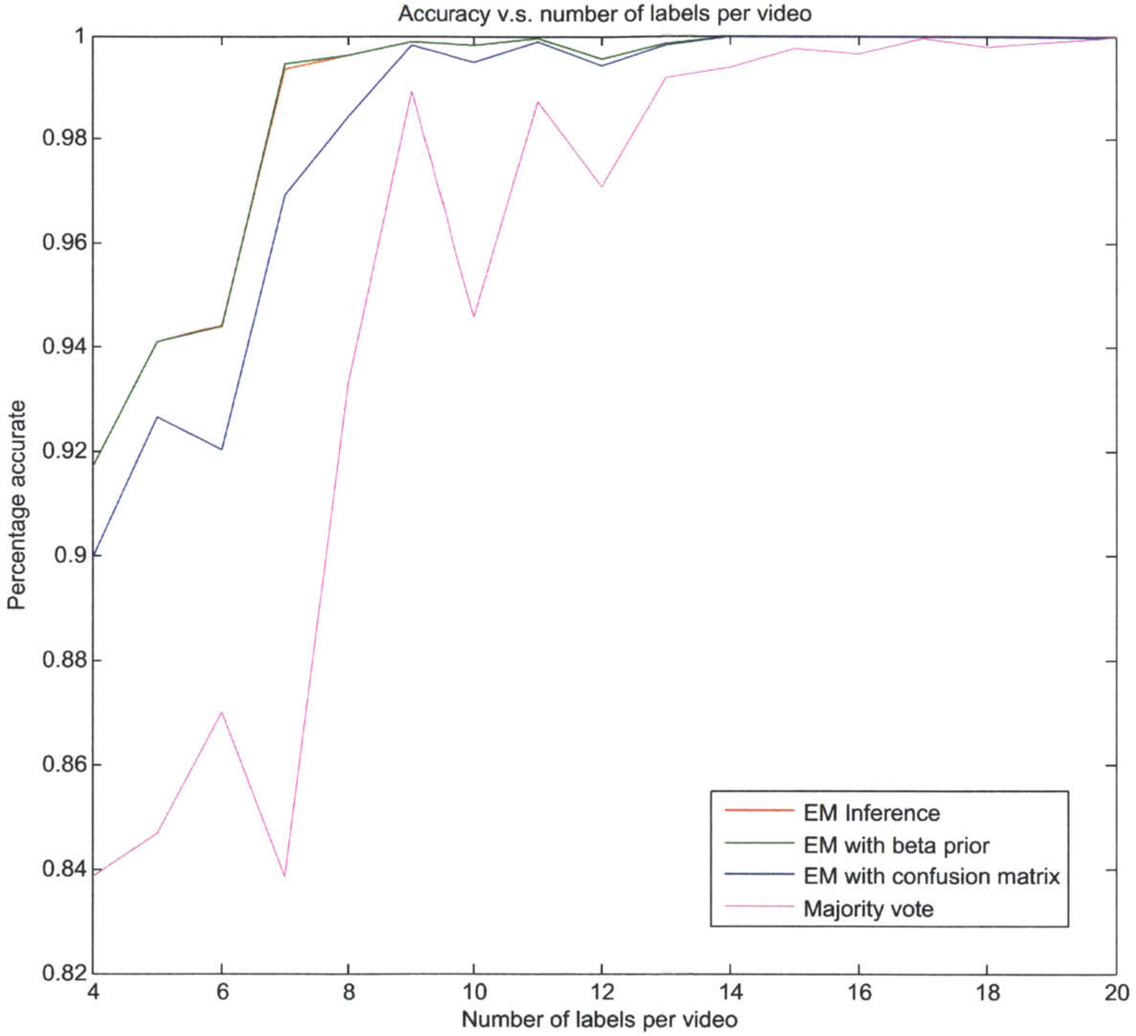


Figure 3-7: Multi-valued annotation simulation result(labelers' accuracy variance 0.4): accuracy of algorithms v.s. number of labels per video

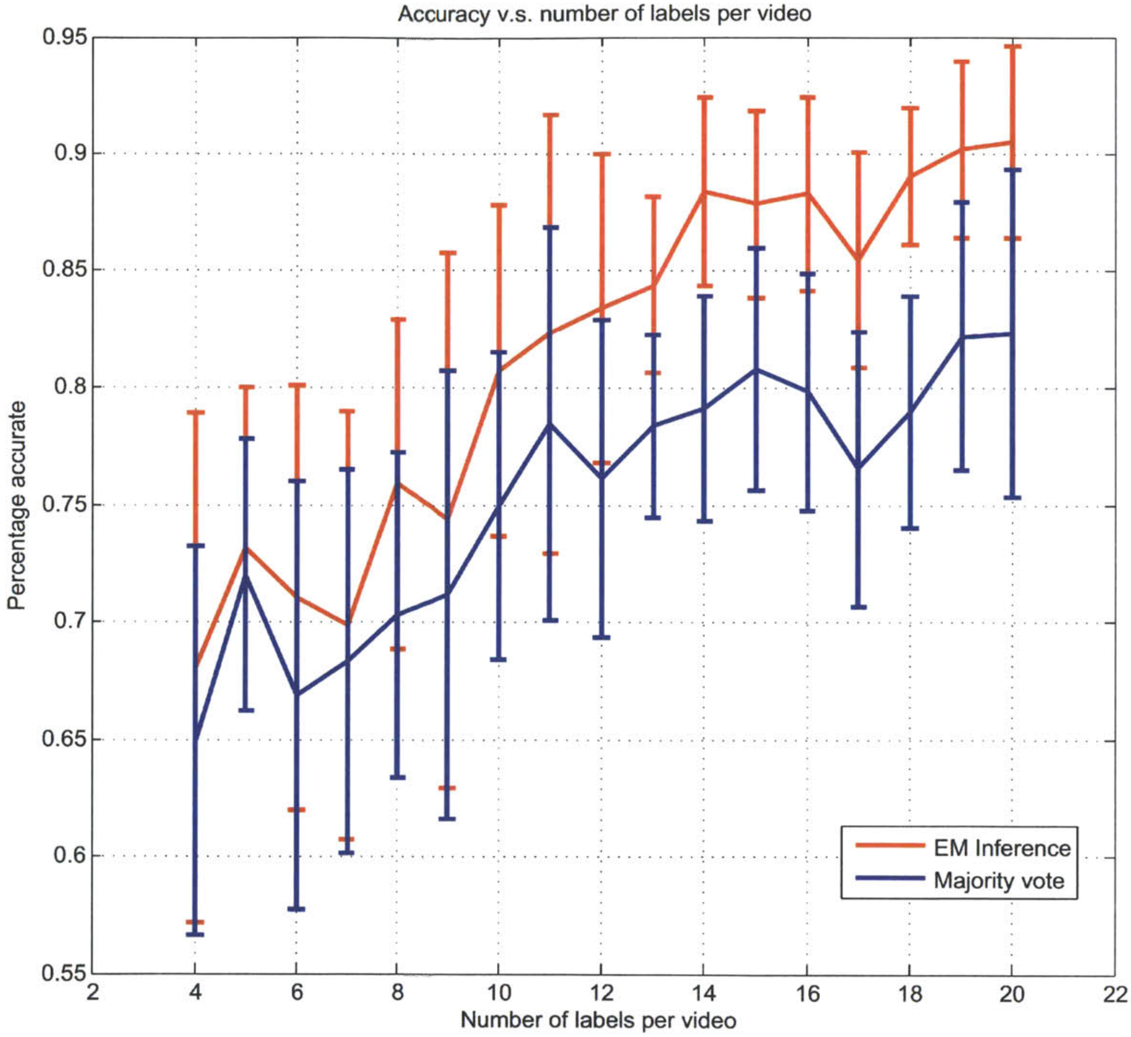


Figure 3-8: Stability of the algorithms: accuracy with standard deviation v.s. number of labels per video

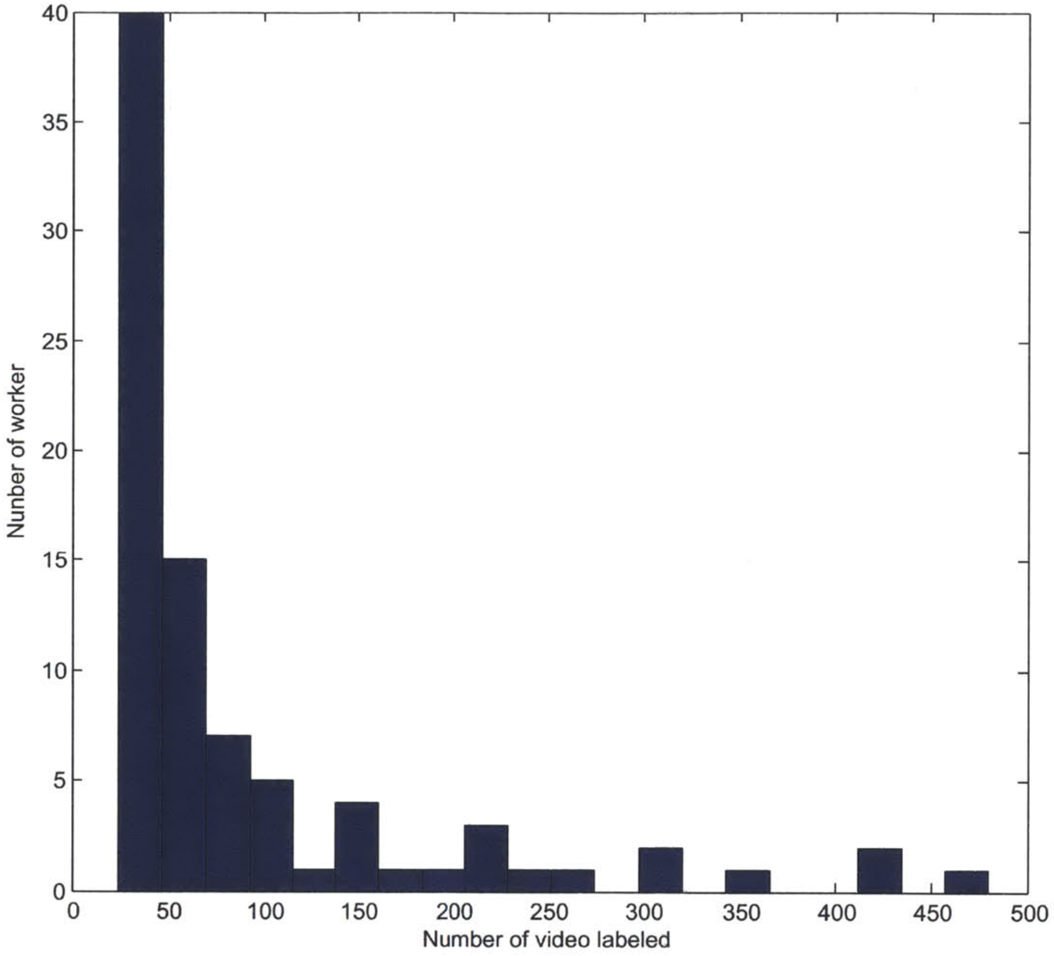


Figure 3-9: Video Clip Labeling Experiment: histogram of the number of videos labeled by each labelers

video clip is annotated by 15 different labelers; in total there are 7200 labels.The histogram of number of video clips labeled by each labelers is shown in Figure 3-9. We can see that most of labelers annotates less than 100 videos; while there is a worker manage to labels all the video clips.

Using the labels obtained from the Mechanical Turk, we infer the video labels using either our algorithms and the majority vote heuristic, and then compare them to the groundtruth. The result is shown in Table 3.1. Our algorithms achieve better performance than majority vote. Notably, EM with confusion matrix obtain the best accuracy 0.9458 among the four algorithms. To understand how the number of labels

Method	EM Inference (s)	EM with beta prior	EM with confusion matrix	Majority vote
Accuracy	0.9375	0.9375	**0.9458**	0.9042

Table 3.1: Video Clip Labeling Experiment: the accuracy of the algorithms

per videos affects the accuracy, we randomly sample m labels from the 15 labels for each video without replacement, where $m = 3, \ldots, 15$ and compute the accuracy of all the algorithms for m. The above procedure is carried out 10 times to take the average. The result plot is shown in Figure 3-10. The EM infernce and ME with beta prior consistently outperform majority vote. The EM with confusion matrix method, which achieves the best accuracy at the end, does the worst at the beginning.

3.8 Moving forward

The label aggregation algorithms we develop based on probabilistic model and EM show promising results in both synthetic data and actual video clips labeling experiment on Mechanical Turk. We would like to leverage the methods to produce labels for long videos. However, there is still one more challenge:

(1) How can we divide a long video into a sequence of video clips, each of which contains a single well-define behavior?

We are going to discuss the above topics in the next chapter.

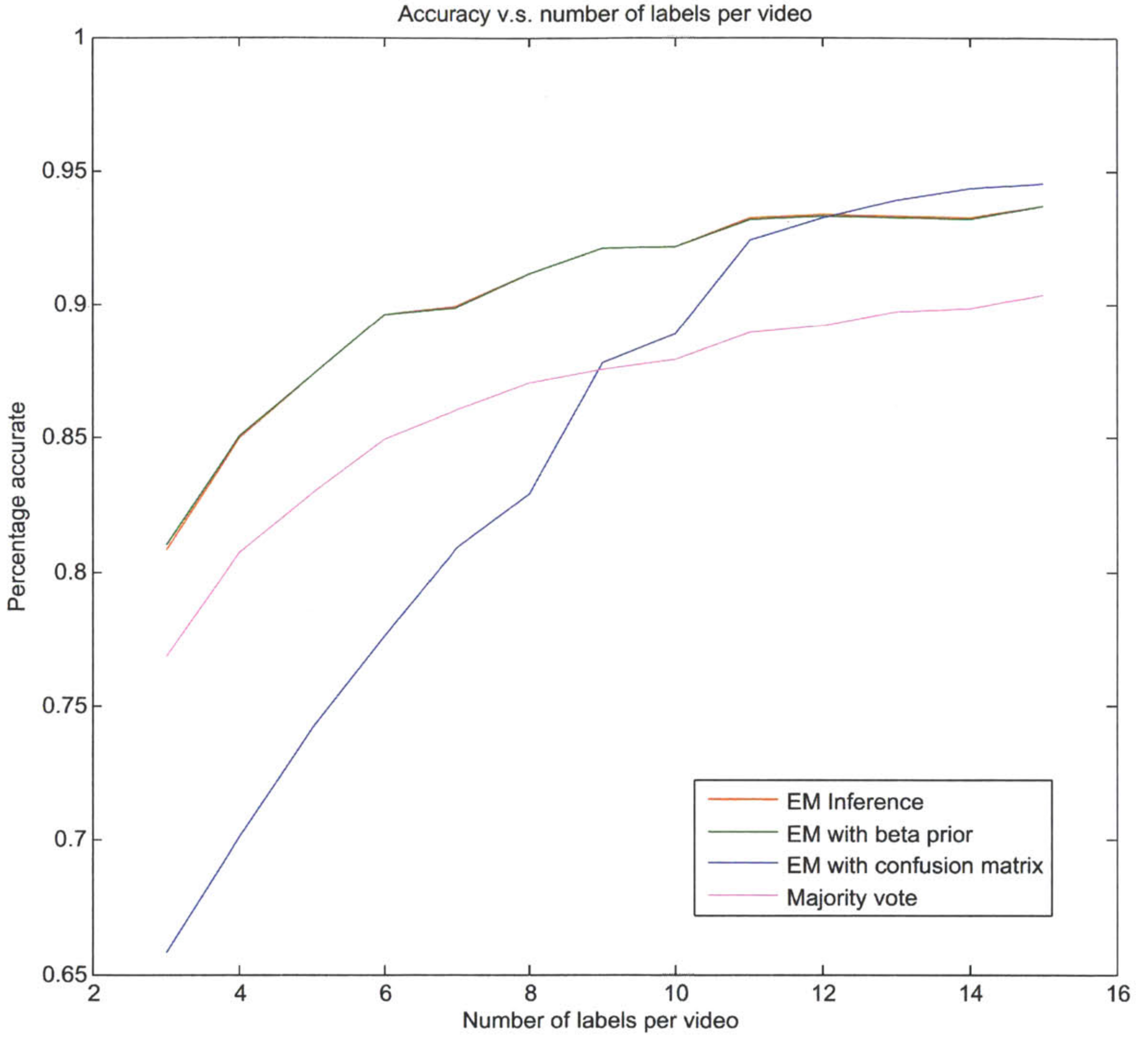


Figure 3-10: Video Clip Labeling Experiment: accuracy v.s. number of labels per video

Chapter 4

Moving towards long video annotation

We have presented the methods to crowdsource labels for short video clips in the previous chapter. In this chapter, we discuss possible ways to automatically label a long video using crowdsourcing platform. The overview of our proposal is shown in Figure 4-1. The system first break long video into a sequence of short behaviorial clips (a clip that only contains a single behavior). The behavioral clips are annotated using our clip-based labeling tools discussed in Chapter 2. Then we use our aggregation algorithms to predict the groundtruth label for each behavioral clip as discussed in Chapter 3. Finally, we combine and smooth the labels of behavioral clips to produce the a fully annotated video. The chapter focuses on the discussion of video segmentation step and the overall system performance.

4.1 Video Segmentation

We can break the long video into small pieces either uniformly or using action segmentation algorithms. To generate video clips uniformly, we need to decide the length of the video clips. After some experiments, we decide that 3 seconds (100 frames) is suitable length. Neither it is too long that each clips often contains more than one behaviors, nor it is too short that result clips contain partial behaviors.

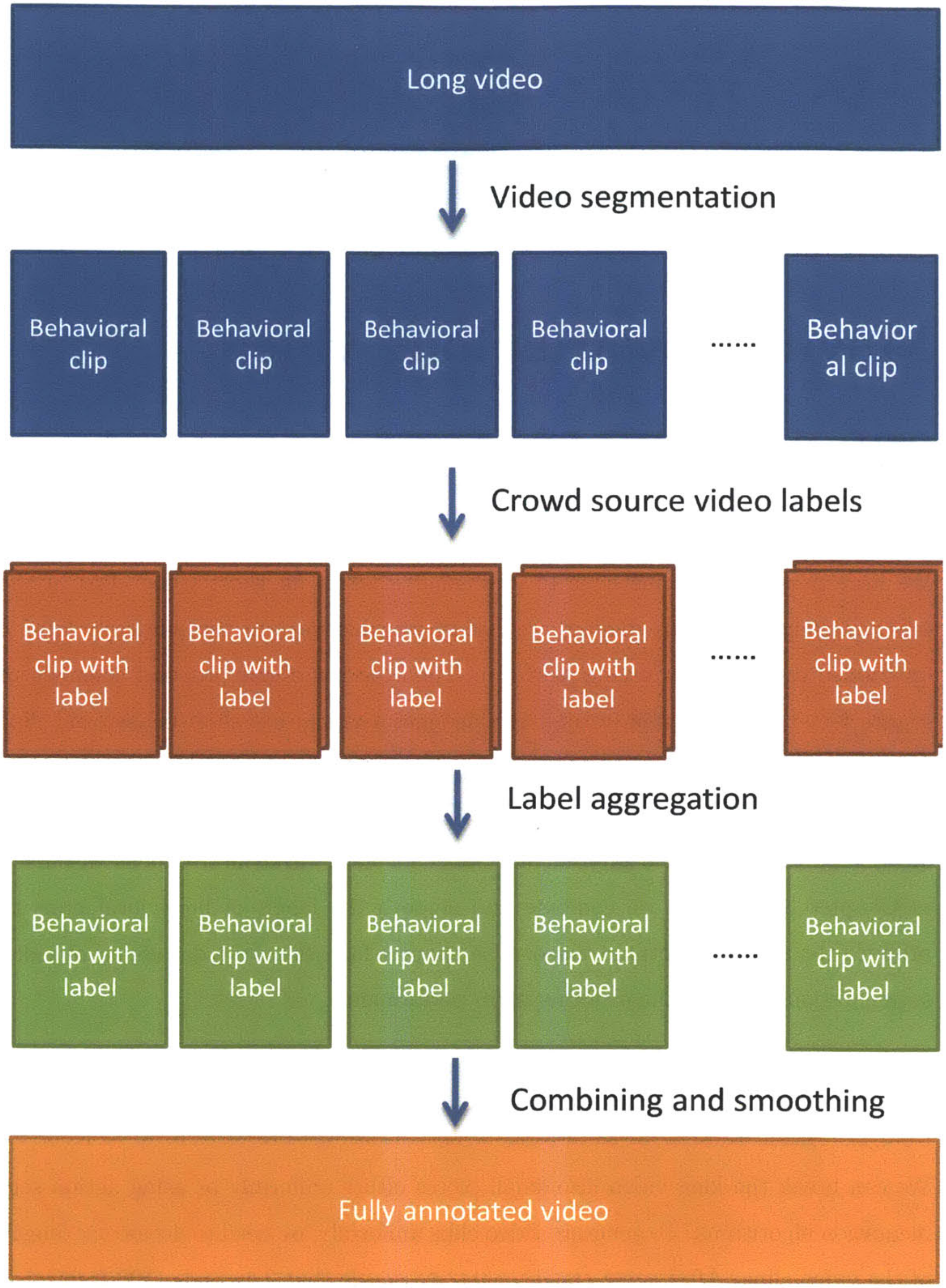

Long video
Video segmentation
Behavioral clip
Behavioral clip
Behavioral clip
Behavioral clip
......
Behavior al clip
Crowd source video labels
Behavioral clip with label
Behavioral clip with label
Behavioral clip with label
Behavioral clip with label
......
Behavioral clip with label
Label aggregation
Behavioral clip with label
Behavioral clip with label
Behavioral clip with label
Behavioral clip with label
......
Behavioral clip with label
Combining and smoothing
Fully annotated video

Figure 4-1: System overview of long video annotation

Since different behavioral clips usually have different length, uniform video segmentation may not give the most desirable results. Instead, we make use of mouse behavioral recognition system developed by Hueihan Jhuang[31] to provide better segmentation. The system is developed from a computational model of motion processing in the primate visual cortex consists of two modules: (1) a feature computation module, and (2) a classification module. We can use the system to classify the video frames into 8 behaviors and group the adjacent frames with same behavior into behavior clips. Moreover, we further make sure every behavior clip is not more than 3 seconds in length.

4.2 Performance of the system

4.2.1 Compare our system with traditional human annotation by hiring college students

To evaluate the performance of our crowdsourcing mouse behavior recognition system, we compare its annotation accuracy to that of human labelers using traditional method. We hire 6 graduate students in MIT to annotate a 10 minutes video using the Flash labeling tools we described in Chapter 2. At the same time, we also annotate the video using our crowdsourcing system. The results are shown in the Table 4.1. Note that for the crowdsourcing system, every video is labeled 10 times by different labelers. As we can see, our crowdsouring system achieves comparable performance as university students. We also plot the change of accuracy of our system as we increase the number of labels per video (i.e. the number of labelers annotate each video) in Figure 4-2. As indicated in the figure, we can improve the system accuracy by increasing the number of labelers for each video.

Now we compare the cost of annotate video by our crowdsourcing system and by hiring university students. The standard rate for hiring student to do the lab work is between \$10 between \$20. And it takes in average 34 minutes for students to annotate the 10 minute video. So it costs about \$6 – \$12 to get the 10 minutes video annotate

Subject	A	B	C	D	E
Accuracy	0.638	0.650	0.691	0.470	0.631
Subject	F	Average(Students)	CSMV	CSEM	CSEMP
Accuracy	0.699	0.630	0.626	0.648	0.648

Table 4.1: Comparing the performance of the crowdsourcing system to that of university students: A - F represent the 6 student annotators; CSMV, CSEM, CSEMP represent respectively our **C**rowdsourcing **S**ystem using **M**ajority **V**ote, **EM** inference and **EM** inference with beta **P**rior.

by one student. On the other hand, our crowdsourcing system pays online workers \$0.12 (it is very generous offer as compared to others on Mechanical Turk) to label 30 short video clips. In total, it only cost about \$0.72 to annotate all video once and only \$7.2 percent to annotate the video 10 times. Consider the accuracy and cost trade-off, to achieve 0.630 accuracy (average student accuracy), our system costs less than \$4.32 as compared to \$6 – \$12 for hiring students. Besides, our system is fully automatic, by which videos can be annotated within hours.

4.2.2 Uniform segmentation v.s. segmentation with behavior recognition algorithm

We would like to know how much does the segmentation methods affect the overall performance of the system. In this experiment, we use the system to annotate a 38 minutes video twice. In the first trial, the video is uniformly segmented into behavioral clips with 3 seconds length. In the second time, the video is segmented using the mouse behavior recognition system from [31]. In each trial, every video is annotated 10 times by different labelers. The experiments results are shown in Table 4.2. The overall accuracy of the system does not change much with different segmentation methods. The uniform segmentation even achieves better accuracy in this case. This agrees with our assumption that if we make the clip short enough, each clip mostly contains 1 behavior.

Besides, we plot the system performance against the number of labels per video as shown in Figure 4-3 and Figure 4-4. As can be expected, the system performance improves as the number of the labels for each video increases.

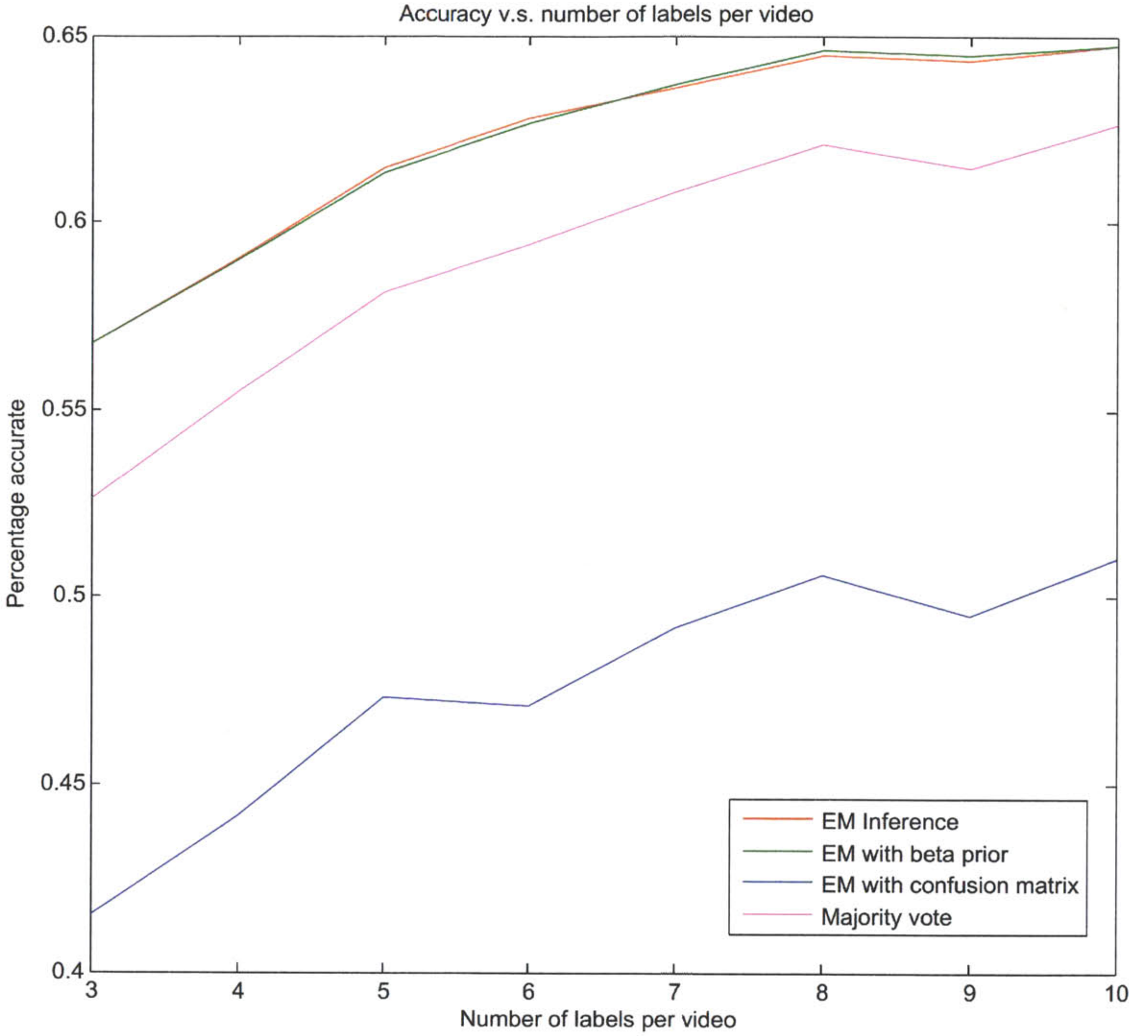


Figure 4-2: System performance on 10-min video annotation

Trial	# clips	# workers	# labels per clip	Accuracy (EM/EM prior/Majority Vote)
Trial 1	688	45	10	0.680/0.680/0.659
Trail 2	937	65	10	0.664/0.665/0.638

Table 4.2: The experiment results of the two trials.

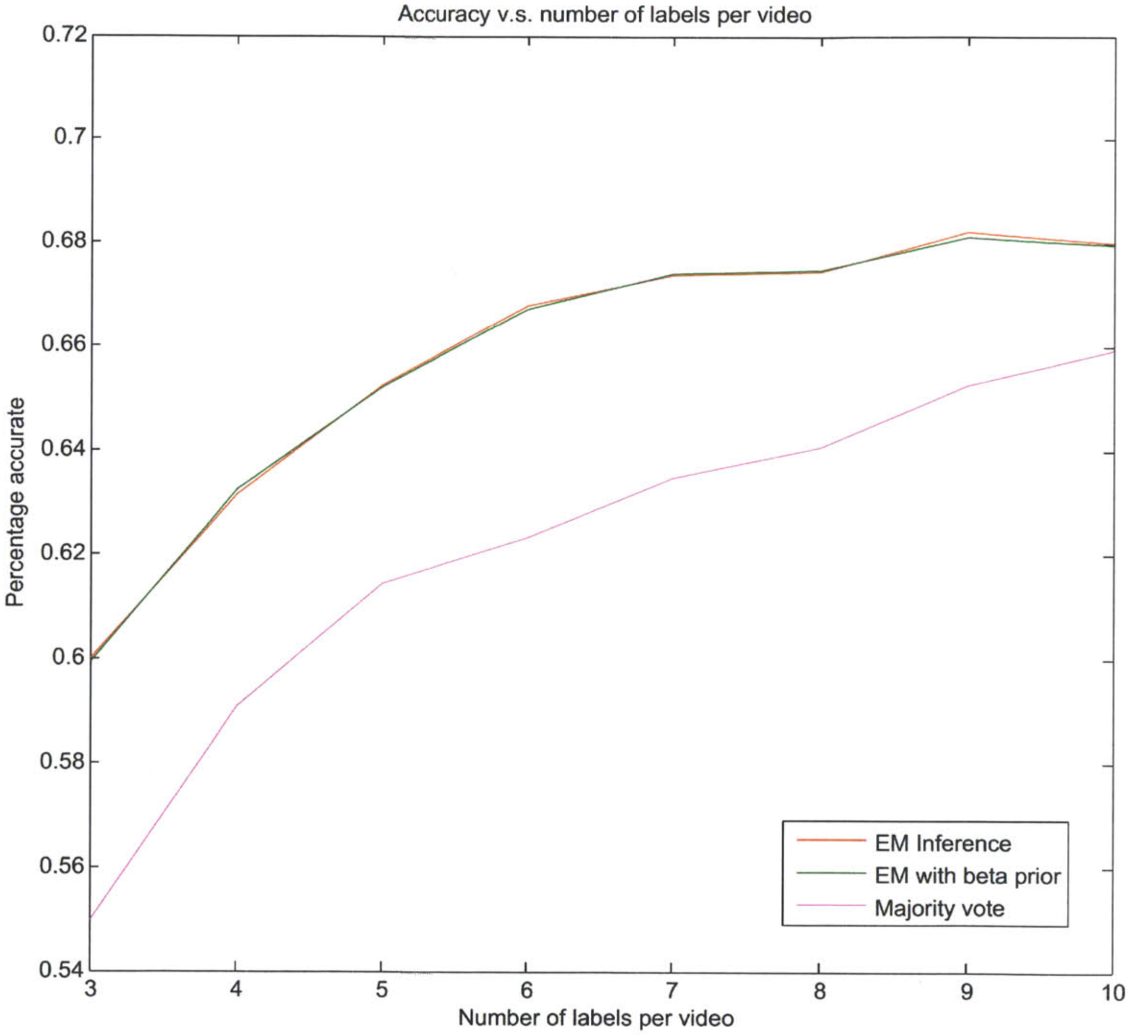


Figure 4-3: System performance on 38-min video annotation using uniform segmentation

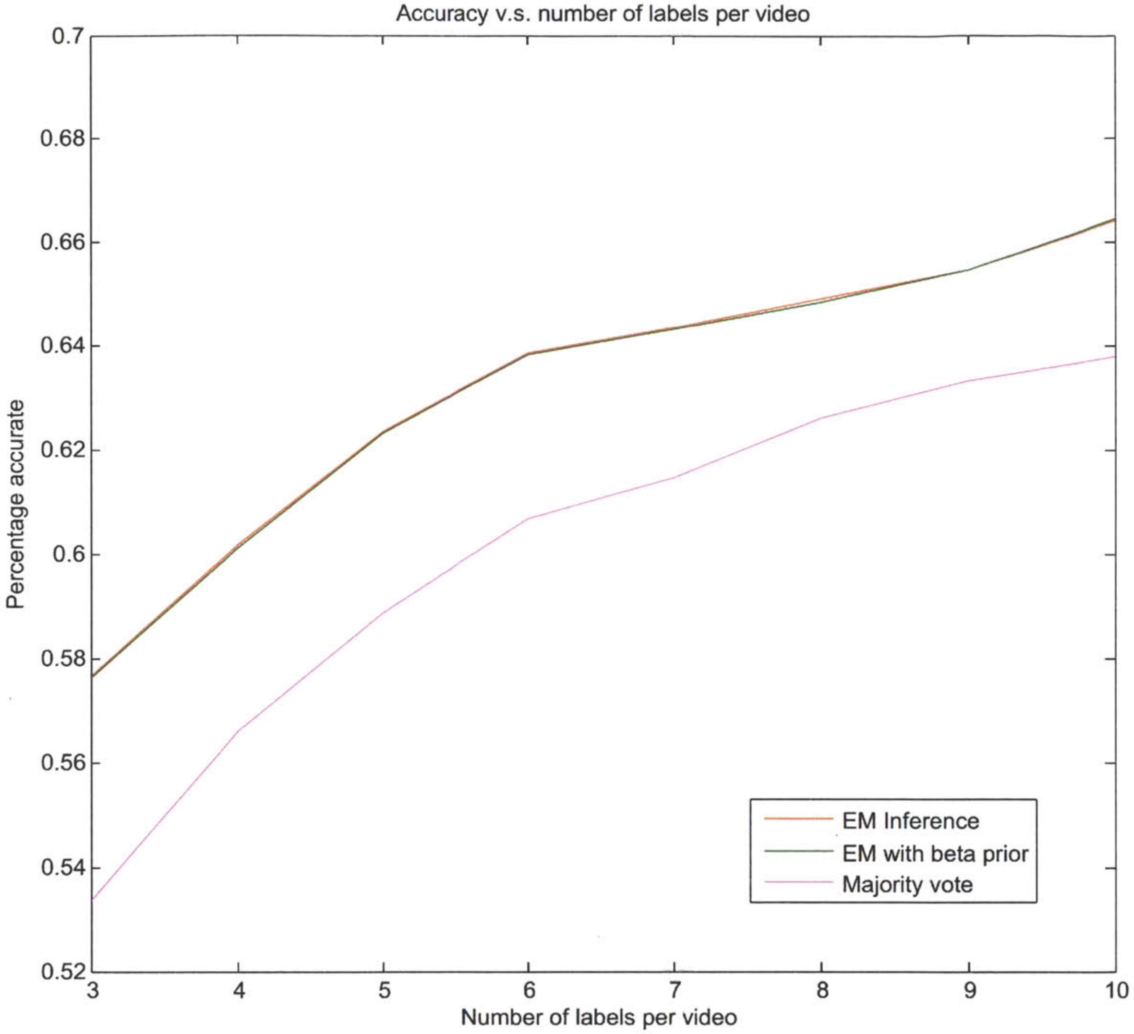


Figure 4-4: System performance on 38-min video annotation using behavior recognition algorithm for segmentation

Chapter 5

Conclusion

Thanks to the advances in artificial intelligence, especially in computer vision and machine learning, machines have been successful in many recognition tasks such as face detection and image search. However, in many critical fields as biological study and medical research, we still cannot completely rely on machine for recognition and detection. Human-based solutions are still predominant in those areas. Therefore, crowdsourcing platforms provide us a cost-efficient solution to those tasks. In this thesis, we have demonstrated an effective crowdsourcing system for mouse behavior recognition, which includes a novel clip-based video labeling tool and an efficient probabilistic aggregation mechanism to predict the true labels from multiple annotations.

Bibliography

[1] Amazon mechanical turk. https://www.mturk.com/mturk/, 2013.

[2] The apache software foundation. http://www.apache.org/, 2013.

[3] Mysql. http://www.mysql.com/, 2013.

[4] M O S Buma A J Spink, R A J Tegelenbosch and L P J J Noldus. The ethovision video tracking system-a tool for behavioral phenotyping of transgenic mice. *Physiology and Behavior*, 73:719–730, 2001.

[5] N. M. Laird A. P. Dempster and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[6] J. Rombouts T.F. Meert A.A.H.P. Megens, J. Voeten and C.J.E. Niemegeers. Behavioural activity of rats measured by a new method based on the piezoelectric principle. *Psychopharmacology*, 93(2):382–388, 1987.

[7] Salesin D Seitz S Agarwala A, Hertzmann A. Keyframe-based tracking for rotoscoping and animation. *ACM Transactions on Graphics (TOG)*, 23:584–591, 2004.

[8] Fleuret F Ali K, Hasler D. Flowboost – appearance learning from sparsely annotated video. *IEEE Computer Vision and Pattern Recognition*, 2011.

[9] Rama Chellappa Ashok Veeraraghavan and Mandyam Srinivasan. Shape-andbehavior encoded tracking of bee dances. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):463–76, 2008.

[10] Johan Auwerx and At El. The European dimension for the mouse genome mutagenesis program. *Nature Genetics*, 36(9):925–927, 2006.

[11] K Branson and S Belongie. Tracking multiple mouse contours (without too many samples). *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[12] K Branson and S Belongie. Tracking multiple mouse contours (without too many samples. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[13] Fitzgibbon A Buchanan A. Interactive feature tracking using kd trees and dynamic programming. *CVPR*, 1:626–633, 2006.

[14] C. J. Taylor C. J. Twining and P. Courtney. Robust tracking and posture description for laboratory rodents using active shape models. *Behavior Research Methods, Instruments, Computers*, 33(3):381–391, 2001.

[15] Deva Ramanan Carl Vondrick, Donald Patterson. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 2012.

[16] Huber D. Personal communication. 2011.

[17] L. Barrington D. Turnbull, R. Liu and G. Lanckriet. A game-based approach for collecting semantic annotations of music. *In 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[18] N Dalal and B Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[19] A.P. Dawid and A.M. Skene. Maximum likelihood estimation of observer errorrates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.

[20] Dong W. Socher R. Li L. Li K. Fei-Fei L Deng, J. Imagenet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–719, 2009.

[21] S. Oh D.R. Karger and D. Shah. Iterative learning for reliable crowdsourcing systems. *Neural Information Processing Systems*, 2001.

[22] Hoiem D Forsyth D Endres I, Farhadi A. The benefits and challenges of collecting richer object annotations. *CVPR Workshop on Advancing Computer Vision with Humans in the Loop*, 2010.

[23] Punita Juneja Adrienne W Mackay-Jennifer MWade Evan H Goulding, A Katrin Schenk and Laurence H Tecott. A robust automated system elucidates mouse home cage behavioral structure. *National Academy of Sciences*, 105(52), 2008.

[24] Williams CKI Winn J Zisserman A Everingham M, Van Gool L. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[25] Barbara Shukitt-hale Gemma Casadesus and James A Joseph. Automated measurement of age-related changes in the locomotor response to environmental novelty and home-cage activity. *Mechanisms of Ageing and Development*, 122:1887–1897, 2001.

[26] Alexei L Vyssotski Michele Angelo Di Bari-Romolo Nonno Umberto Agrimi Giacomo DellOmo, Elisabetta Vannoni and Hans peter Lipp. Early behavioural changes in mice infected with bse and scrapie: automated home cage monitoring

reveals prion strain differences. *European Journal of Neuroscience*, 16:735–742, 2002.

[27] M A Giese and T Poggio. Neural mechanisms for the recognition of biological movements. *Nature Review Neuroscience*, 4:179–192, 2003.

[28] L Wolf H Jhuang, T Serre and T Poggio. A biologically inspired system for action recognition. *ICCV*, 2007.

[29] N. Edelman T. Poggio T. Serre H. Jhuang, E. Garrote. Vision-based automated home cage behavioral phenotyping. *the 7th International Conference on Methods and Techniques in Behavioral Research*, 2010.

[30] T. Serre H. Jhuang and T. Poggio. Computational mechanisms for the motion processing in visual area mt. *Neuroscience Meeting Planner, San Diego, CA: Society for Neuroscience*, 2010.

[31] X. Yu V. Khilnani T. Poggio A Steele H. Jhuang, E. Garrote and T Serre. Automated home-cage behavioral phenotyping of mice. *Nature communications*, 2010.

[32] Eric D Hoopfer David J Anderson Heiko Dankert, Liming Wang and Pietro Perona. Automated monitoring and analysis of social behavior in drosophila. *Nature Methods*, 6(4), 2009.

[33] Thomas C Henderson and Xinwei Xue. Constructing comprehensive behaviors: A simulation study. *International Conference on Computer Applications in Industry and Engineering*, 2005.

[34] T Wu J Bergsma J Whitehill, P Ruvolo and J Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Neural Information Processing Systems*, pages 2035–2043, 2009.

[35] Xianhua Jiang Jane Brooks Zurn and Yuichi Motai. Video-based tracking and incremental learning applied to rodent behavioral activity under near-infrared illumination. *Behavior Research Methods, Instruments, Computers*, 56(6):2804–2813, 2007.

[36] R. Jin and Z. Ghahramani. Learning with multiple labels. *Neural Information Processing Systems*, pages 921–928, 2003.

[37] D. Koller and N. Friedman. Probabilistic graphical models: principles and techniques. *MIT press*, 2009.

[38] John Bender Pietro Perona Kristin Branson, Alice A Robie and Michael H Dickinson. High-throughput ethomics in large groups of drosophila. *Nature Methods*, 6(6), 2009.

[39] A J Spink L P Noldus and R A Tegelenbosch. Ethovision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments, Computers*, 33(3):398–414.

[40] A J Spink L P Noldus and R A Tegelenbosch. Ethovision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments, Computers*, 33(3), 2001.

[41] C. McMillen D. Abraham L. von Ahn, B. Maurer and M. Blum. re-captcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465, 2008.

[42] Schmid C Rozenfeld B Laptev I, Marszalek M. Learning realistic human actions from movies. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[43] Philipp Kegel Mareike Kritzler Antonio Kruger Lars Lewejohann, Anne Marie Hoppmann and Norbert Sachser. Behavioral phenotyping of a murine model of alzheimers disease in a seminaturalistic environment using rfid tracking. *Behavior Research Methods*, 41:850–856, 2009.

[44] Doermann D Mihalcik D. The design and implementation of viper. *Technical Report*, 2003.

[45] Dong J. Feng J. Yan S. Ni, Y. Purposive hidden-object-game: embedding human computation in popular game. *ACM MM*, 2011.

[46] G Cottrell P Dollar, V Rabaud and S Belongie. Behavior recognition via sparse spatio-temporal features. *In visual surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[47] H. Sigg P. Tamborini and Zbinden G. Quantitative analysis of rat activity in the home cage by infrared monitoring. application to the acute toxicity testing of acetanilide and phenylmercuric acetate. *Archives of Toxicology*, 63:85–96, 1989.

[48] S. Belongie P. Welinder, S. Branson and P. Perona. The multidimensional wisdom of crowds. *Neural Information Processing Systems*, 2010.

[49] S. Belongie P. Welinder, S. Branson and P. Perona. Online crowdsourcing: rating annotators and obtaining costeffective labels. *Neural Information Processing Systems*, 2010.

[50] J.B.I Rousseau P.B.A Van Lochem, M.O.S Buma and L.P.J.J Noldus. Automatic recognition of behavioral patterns of rats using video imaging and statistical classification. *Measuring Behavior*, 1998.

[51] J. Peng Q. Liu and A. Ihler. Variational inference for crowdsourcing. *Neural Information Processing Systems*, 2012.

[52] Fisher R. The pets04 surveillance ground-truth data sets. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 1:1–5, 2004.

[53] Kakade S Ramanan D, Baker S. Leveraging archival video for building face datasets. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[54] Torralba A. Murphy K. Freeman W. Russell, B. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

[55] V. Rabaud S. Steinbach and S. Belongie. Soylent grid: its made of people ! *In International Conference on Computer Vision*, 2007.

[56] Kraaij W Smeaton A, Over P. Evaluation campaigns and trecvid. *ACM international workshop on Multimedia information retrieval*, pages 321–330, 2006.

[57] Andrew D Straw Steven N Fry, Nicola Rohrseitz and Michael H Dickinson. Trackfly: virtual reality for a behavioral system analysis in free-ying fruit flies. *Journal of Neuroscience Methods*, 171:110–117, 2008.

[58] E. Garrote T. Poggio T.* Serre, H.* Jhuang and Steele A. Automatic recognition of rodent behavior: A tool for systematic phenotypic analysis. 2009.

[59] Xiangdong Tang and Larry D Sanford. Home cage activity and activity-based measures of anxiety in 129p3/j, 129x1/svj and c57bl/6j mice. *Physiology and behavior*, 84(1):105–15, 2005.

[60] Adrian Hilton Thomas B Moeslund and Volker Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[61] Jean-Marie Aerts Rudi DHooge Toon Leroy, Stijn Stroobants and Daniel Berckmans. Automatic analysis of altered gait in arylsulphatase a-deficient mice in the open field. *Behavior Research Methods, Instruments, Computers*, 41:787–794, 2009.

[62] L.H. Zhao G.H. Valadez C. Florin L. Bogoni V.C. Raykar, S. Yu and L. Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.

[63] Paul Viola and Michael Jones. Robust real-time object detection. *IEEE International Conference on Computer Vision*, 2001.

[64] Hays J Vittayakorn S. Quality assessment for crowdsourced object annotations. *the British Machine Vision Conference*, 2011.

[65] Blum M Von Ahn L, Liu R. Peekaboom: a game for locating objects in images. *the SIGCHI conference on Human factors in computing systems, ACM*, pages 55–64, 2006.

[66] Dabbish L Von Ahn L. Labeling images with a computer game. *the SIGCHI conference on Human factors in computing systems, ACM*, pages 319–326, 2004.

[67] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1).

[68] Y. Weiss and W.T. Freeman. On the optimality of solutions of the max-product beliefpropagation algorithm in arbitrary graphs. *Information Theory, IEEE Transactions on*, 47(2):736–744, 2001.

[69] D.Lin D. Anderson X. Burgos-Artizzu, P. Dollar and P.Perona. Social behavior recognition in continuous videos. *CVPR*, 2012.

[70] Xinwei Xue and Thomas C Henderson. Video based animal behavior analysis from multiple cameras. *International Conference On Multisensor Fusion and Integration for Intelligent Systems*, pages 335–340, 2006.

[71] G. Fung M. Schmidt G. Hermosillo L. Bogoni L. Moy Y. Yan, R. Rosales and J. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. *The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, pages 932–939, 2010.

[72] Liu C Torralba A Yuen J, Russell B. Labelme video: Building a video database with human annotations. *International Conference of Computer Vision*, 2009.

[73] Tucker Balch Zia Khan and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence*, 27:1805–1819, 2005.