Learning and Recognizing Activities in Streams of Video

Dinesh Govindaraju and Manuela Veloso

Computer Science Department
Carnegie Mellon University
dineshg@cmu.edu, veloso@cs.cmu.edu

Abstract

This paper presents an algorithm for learning the underlying models which generate streams of observations, found in video data, which encode activities performed by a person who appears in the video. With these learned models, we then aim to carry out recognition in new video streams which display the same activities as the ones that were learned. Our algorithm represents the underlying models as regular Hidden Markov Models as the problem includes sequential and temporally discrete observations and uses the Baum Welch algorithm in learning the underlying models.

Introduction

Effective behavior recognition using a Camera Assisted Meeting Event Observer [1] in the context of an office meeting is a challenging problem with many potential benefits to be reaped. If when given a video sequence of an office meeting, we are able to accurately segment the video and recognize consistent activities of meeting attendees, we would then be able to augment higher level functionality such as recognizing behaviors and with these, infer which attendees are presenting agenda items and even gauge attendee interest levels during the meeting.

Previous work done in activity recognition includes using optical flow [8, 12], classification hierarchies [14], other model-based methods [15, 16] as well as variants of HMMs [7] to carry out the recognition. Instances where regular HMMs have been used to carry out activity recognition include cases where the structure of the models are already known [6] or have been generated using implicit information possessed by the researcher regarding the problem [4, 9, 10, 11] such as the number of states.

In this paper we are interested in extending earlier work done in [13] by learning the underlying activity models automatically from video data with which we can subsequently carry out activity recognition. We wish to make the learning process automatic so as to make it feasible in carrying out recognition on video streams containing a large number of activities. Generating the underlying models by hand in cases where there are many activities would be time consuming to the extent of rendering the process impractical. Automating the process also allows us to do learning on datasets which are very general. This has the advantage of making the process versatile as the learning is applicable to a wide variety of data and is not constrained by particular aspects of a specific problem.

This paper is structured as follows. In the next section we define the problem in more detail and identify some specific problems that arise when generating a solution. We then outline the approach taken by our algorithm and present some results by evaluating it against a hand labeled control case. We finally discuss the results of the evaluation and some possible future improvements to the algorithm.

Problem Definition

In this paper we focus on the problem of activity recognition given a video stream showing a person engaged in a set of activities. In particular, we will first run face detection on the sequence to obtain the x and y coordinates of the person's face in each frame. We will then generate a stream of movement deltas along the x and y coordinates by taking the difference of the detected face coordinates between adjacent frames. This sequence of movement deltas will then be the features we will use in carrying out the activity recognition (where each pair of x and y delta values is a specific emission in our observation space).

Given the movements deltas for the video stream and a few training segments of the various activities the person is engaged in (also expressed in terms of x and y deltas), we wish to be able to accurately recognize the correct activity the person is engaged in for each image in the unlabeled sequence. We want to be able to do this by first learning the parameters of some underlying model for each of the activities using the training segments and subsequently using these learned models to carry out activity recognition in the image sequence.

One issue that arises when trying to solve this problem is that we don't want to make any assumptions on the segment length of each activity in the unlabeled image sequence and so cannot segment the sequence automatically. Another problem that surfaces is that we do not have any implicit knowledge about the activities that

we wish to recognize and so cannot make any assumptions about the underlying models such as the number of states generating the observations.

Approach

Our approach involves two stages. In the first, we use the training data to learn the underlying models and in the second, we use the learned models to recognize activities in the unlabeled image sequence.

Learning Underlying Models

We start our approach to the problem by generating the underlying models for each activity that we will later use for recognition. At this point we will make the assumption that the underlying models can be closely approximated by using regular Hidden Markov Models [3] consisting of (1) as the problem includes sequential and temporally discrete observations.

$$
\begin{aligned}
&N = \{s_i\} && \text{- the states in the model}\\
&M = \{o_i\} && \text{- the observation space}\\
&A = \{a_{ij}\} && \text{- the state transition matrix, where}\\
&\qquad a_{ij} = P(S_{t+1} = s_j \mid S_t = s_i), \quad 1 \le i, j \le |N| &&\\
&B = \{b_i(o_k)\} && \text{- the observation probabilities, where}\\
&\qquad b_i(o) = P(o \mid S_t = s_i), \quad 1 \le i \le |N| &&\\
&\pi = \{\pi_i\} && \text{- the initial state distribution, where}\\
&\qquad \pi_i = P(S_1 = s_i), \quad 1 \le i \le |N| &&
\end{aligned}
\tag{1}
$$

We will first hand label sets of training segments for each of the activities we are learning. The training segments for each activity are streams of observations showing the activity being performed once through and can be of varying lengths. We then run the Baum Welch algorithm [3] to learn the optimal model for each activity using the training segments.

Before Baum Welch can be carried out, we first need to determine the possible space of observations that can be emitted by the learned model and we obtain these observations by running through all the training segments for each activity and including all observations we find. In the case that we find an observation in the unlabeled sequence that we have not encountered before, we augment the learned observation probability matrices by adding the new observation (assigning it with a small non-zero probability) and re-normalizing the matrices.

We also need to find the number of states, Q, which our model will contain before we can commence Baum Welch. We will choose this optimum value of Q by carrying out N-fold cross validation. To do this, we first set the upper bound for Q by choosing the length of the longest training segment, Q_u (a value of Q corresponding to the case where each observation in the longest training segment is emitted by a single state). We then iterate over all values of Q (from 1 to Q_u) and for each, we generate the most likely overall model for the activity we are learning by averaging the parameters of the models generated for each of the training segments as in [2]. We then perform N-fold cross validation on these Q_u number of models using the criterion of how well the model was able to discriminate training segments of the activity we are trying to learn, from the training segments of other activities. In performing the N-fold cross validation, we initially used the log likelihood of generating the held out training segments as the criterion for testing to find the optimal value for Q, but found that this performed slightly worse in practice.

With this optimal value, we then run Baum Welch using different sets of randomly initialized parameters and choose the model with the best sum of log likelihoods of generating all of the activity's training segments. The best performing model, λ_a , will then be the one we use as the underlying model for the activity we were learning.

Activity Recognition

Once we have learned the underlying model for each activity, we are ready to start recognizing activities in the new image sequence of length L. We start by defining a width, w, and from the beginning, sequentially consider the set of windows of observations (2) which are each w frames wide.

$$
\{[o_m, o_{m+1}, \ldots, o_{m+w-1}]\} \qquad \text{for } 1 \le m \le (L - w + 1) \tag{2}
$$

We then take each window as a separate observation segment of length w and calculate the likelihood, (3) for each activity.

$$
P(o_m, o_{m+1}, \ldots, o_{m+w-1} \mid \lambda_a) \tag{3}
$$

We do this by obtaining (4) which is similar to the calculation of likelihood in [3].

$$
\begin{aligned}
&P(o_m, o_{m+1}, \ldots, o_{m+w-1} \mid \lambda_a) = \sum_i \alpha_{m+w-1}(i)\\
&\text{where } \alpha_t(i) \text{ is defined recursively by}\\
&\alpha_{t+1}(j) = \sum_i \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1})\\
&\alpha_m(i) = \pi_i \cdot b_i(o_m)
\end{aligned}
\tag{4}
$$

In practice it was found that a good window width, w, to use was the average of the lengths of the training segments for all activities.

Once we have generated the likelihoods for each activity using its learned model, we then proceed to generate the recognized activity for each frame of the image sequence, *activity(frame_index)*, by assigning the middle frame in each sequential window with the activity that has the highest likelihood and so satisfies (5).

$$\text{activity}\left(t+\lceil w/2\rceil\right) = \arg\max_a \left[P(o_t, o_{t+1}, ..., o_{t+w-1} \mid \lambda_a)\right] \quad (5)$$

For frames whose index is either less than $\lceil w/2\rceil$ or more than $L-\lceil w/2\rceil$ we generate the likelihoods based on the observations of the first and last windows of the image sequence. At the end of the procedure we have the set of labels, *activity(i)*, identifying the recognized activity of each frame, *i*, for the entire image sequence.

Evaluation and Results

Evaluation was carried out on an image sequence consisting of a person engaged in the activities shown in Table 1.

Activity	Description
Stand	Having stood up
Standing	Process of standing up
Sit	Having sat down
Sitting	Process of sitting down
Fidget Left	Process of moving slightly left and stopping
Fidget Right	Process of moving slightly right and stopping

TABLE 1. ACTIVITIES BEING OBSERVED

The image sequence was 87 seconds in duration, consisting of 1296 individual frames with the x and y coordinates of the face of the person engaged in the activities being detected in each frame. This stream of coordinates was then processed to generate the movement deltas along the x and y directions of the face between subsequent frames. This stream of changes in the position coordinates of the face throughout the image sequence was used as the stream of observations. Examples of frames in the image sequence after the face has been detected can be seen in Figure 1.

FIGURE 1. FRAMES WITH DETECTED FACES

The sequence of observations was first hand labeled to indicate the activity each frame displayed. Segments of consecutive observations displaying the same activity were then extracted from the sequence and 4 training segments from each activity were used to learn the underlying model for that activity. These 24 individual training segments accounted for approximately 40% of all the activity found in the entire sequence. Using 4 training segments to carry out the learning was found to be optimal given that we wanted to minimize the amount of human labeling required. Final accuracies of recognizing activities in the new video when varying the number of training segments used to learn the underlying activity models can be seen in Figure 2.

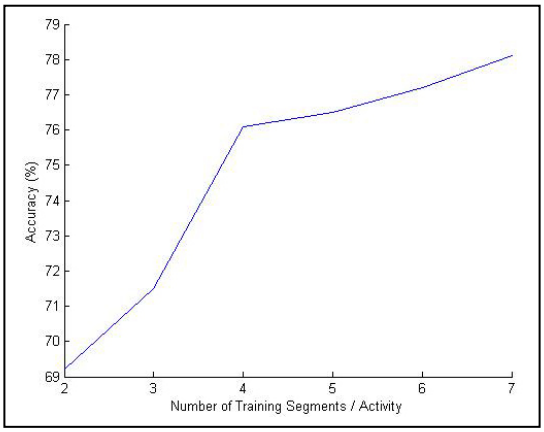

FIGURE 2. ACCURACIES WHEN VARYING THE NUMBER OF TRAINING SEGMENTS

Once the underlying models were learnt for each activity, they were used to compute the likelihood of a sliding window over the new unlabeled image sequence. The likelihoods obtained for 3 of the 6 activities from each window over a portion of the image sequence can be seen in Figure 3. A value of 21 was used for the window width, *w*, as this was the average length of the 24 training segments used to learn the underlying models.

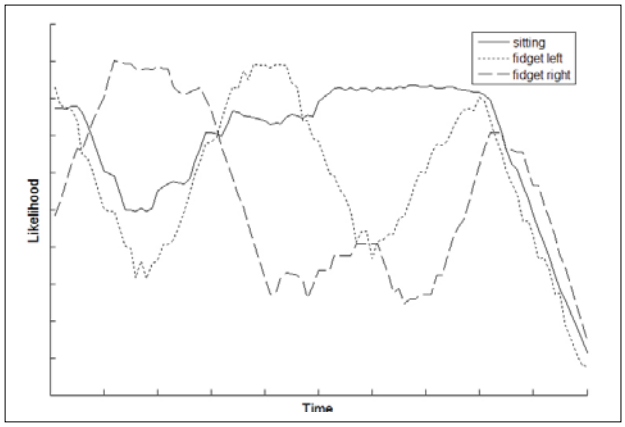

FIGURE 3. OBTAINED LIKELIHOODS

With the determined likelihoods, the image sequence was then assigned with recognized activities by picking the highest likelihood as according to the developed algorithm and comparisons were made against the hand labeled sequence. Figure 4 displays a plot of the activities assigned to frames which were correctly recognized by the developed algorithm and so were identical to the hand assigned activities. Figure 5 displays a plot of the activities assigned by the developed algorithm for the frames whose algorithm recognized activity differed from the hand assigned activity and so were deemed to be incorrect. Figure 6 displays a plot of the hand assigned activities for those frames which were deemed to be incorrect.

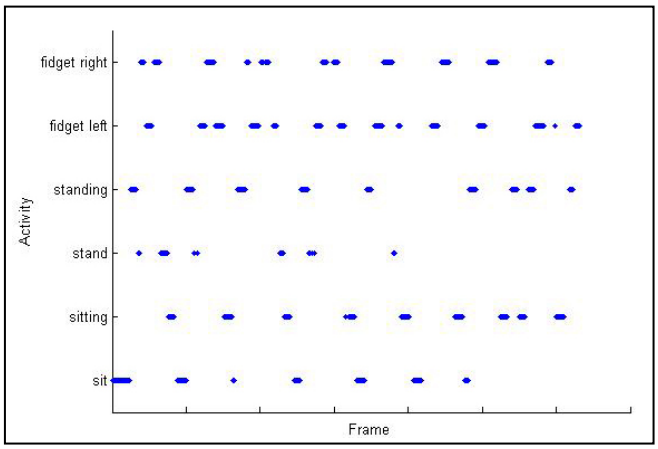

FIGURE 4. ACTIVITIES OF CORRECTLY RECOGNIZED FRAMES

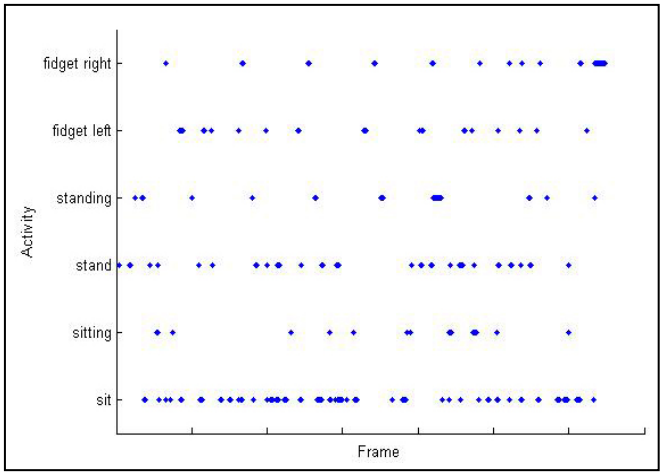

FIGURE 5. ALGORITHM ASSIGNED ACTIVITIES OF INCORRECTLY RECOGNIZED FRAMES

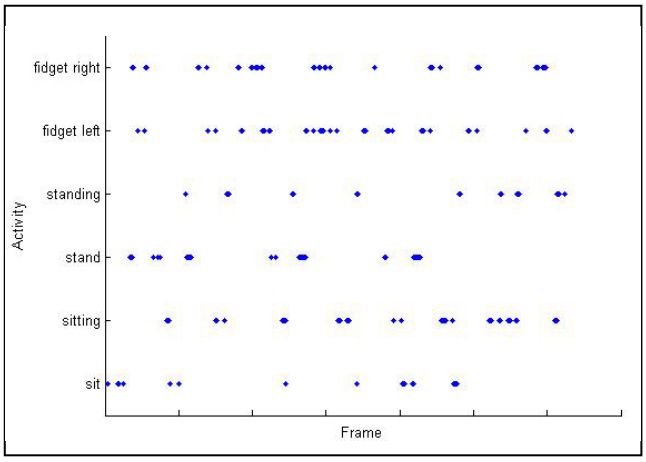

FIGURE 6. HAND ASSIGNED ACTIVITIES OF INCORRECTLY RECOGNIZED FRAMES

From the comparisons, it can be seen that the majority of the misclassifications are of 2 types. The first type occurs when misclassifications are at the edges of distinct activity segments in which they are instead assigned the label of the ‘sit’ activity. The second type of misclassification occurs when the ‘stand’ activity is misclassified as a ‘sit’ activity and vice versa. The first type of misclassification most likely occurs because there is little activity in the frames between distinct activities as there is not much movement when transitioning from one activity to the other. The observation sequence of movement deltas during these transition periods are therefore similar to a sequence emitted when a person is engaged in the ‘sit’ activity. The second type of misclassification occurs for a similar reason in that the movement deltas emitted when a person is engaged in the ‘sit’ and ’stand’ states are similar as in both cases the person is not moving significantly. The overall accuracy of the algorithm was found to be approximately 76% when compared with the hand assigned labels.

Conclusions and Future Work

From the results it was found that the developed algorithm performed fairly well when compared to labeling the activities of the image sequence by hand. When a comparable evaluation was done in previous similar work [13] of which this paper is an extension, activity labeling accuracy was found to be 90.8% when compared to hand labeled ground truth. However, it should be noted that the algorithm presented in this paper has the advantage of being able to automatically learn the underlying activity models whereas models in the previous work were generated by hand. This allows the algorithm to be more versatile as well as making it more practical in cases where the image sequence contains a large number of activities to be recognized.

One possible improvement that could be made to the algorithm to reduce the number of misclassifications would to be to learn an overall model generating the sequence of activity segments themselves. As an example, if such a model was learned for the image sequence presented in the evaluation section, it would consist of 6 states where each state would correspond to one of the activities shown (sit, sitting, stand, standing, fidget left and fidget right). This overall model would therefore generate the sequence of the various activities we observe in our image sequence (where each specific activity would itself consist of a series of observations) and would allow us to deduce that the probability of transitioning from the sitting state to the stand state was very low. Once we have done the initial activity recognition in the unlabeled sequence using the algorithm presented in this paper, we could see if swapping the labeled activities of any frames with the activities of the next highest likelihood would increase the likelihood of observing the entire activity sequence when using this overall model and this might improve labeling accuracy.

When choosing the value of the window width, averaging the lengths of the training segments for all activities might not be optimal especially when the average lengths of training segments for each activity vary by large amounts. A possible improvement to this is to standardize the lengths of the initial training segments by using dynamic time warping [5] and then interpolating between observations in each segment. If all training segments could be normalized to a particular window width, *w*, which we could then use when carrying out activity recognition on the unlabeled image sequence, recognition accuracy might be increased.

References

[1] *P. Rybski, F. De la Torre Frade, R. Patil, C. Vallespi, M. Veloso, and B. Browning.* CAMEO: The Camera Assisted Meeting Event Observer, *Tech. Report CMU-RI-TR-04-07, Robotics Institute, Carnegie Mellon University, January, 2004.*

[2] *Davis, Richard I. A., Lovell, Brian C., Caelli, Terry,* Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences. *International Conference on Pattern Recognition 2002, August 11-14 II, pages 168-171.*

[3] *L Rabinier,* A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings IEEE, 1989.*

[4] *K. Han, M. Veloso,* Automated Robot Behavior Recognition Applied to Robotic Soccer, *In John Hollerbach and Dan Koditschek, editors, Robotics Research: the Ninth International Symposium, pages 199--204. Springer-Verlag, London, 2000.*

[5] *T. Oates, L. Firoiu, P. Cohen,* Using Dynamic Time Warping to Bootstrap HMM-Based Clustering of Time Series. In Sequence Learning: Paradigms, Algorithms and Applications. *Ron Sun and C. L. Giles (Eds.) Springer-Verlag: LNAI 1828. 2001.*

[6] *J. Binder, D. Koller, S. Russell, K. Kanazawa,* Adaptive probabilistic networks with hidden variables. *Machine Learning, 29:213–244, 1997.*

[7] *N. Oliver, E. Horvitz, A. Garg,* Layered Representations for Human Activity Recognition, *ICMI '02.*

[8] *P. Huang, C. Harris, M. Nixon,* Recognizing humans by gait via parametric canonical space. *Artificial Intelligence in Engineering, 13(4):359–366, October 1999.*

[9] *A. Kale, N Cuntoor, V. Krüger,* Gait-Based Recognition of Humans Using Continuous HMMs, *FGR '02.*

[10] *Q. He, C. Debrunner,* Individual Recognition from Periodic Activity Using Hidden Markov Models, *Workshop on Human Motion 2000: 47-52.*

[11] *A. Madabhushi, J. K. Aggarwal,* A Bayesian Approach to Human Activity Recognition, *In Proc. of the 2nd International Workshop on Visual Surveillance.*

[12] *A. Efros, A. Berg, G. Mori, J. Malik,* Recognizing Action at a Distance, *In ICCV 2003.*

[13] *P. E. Rybski, M. Veloso,* Using Sparse Visual Data to Model Human Activities in Meetings, *In the Modeling Other Agents from Observations workshop at AAMAS'04, the 3rd International Joint Conference on Autonomous Agents & Multi-Agent Systems, New York, New York, 2004.*

[14] *C. Stauffer, W.E.L. Grimson,* Learning Patterns of Activity Using Real-Time Tracking, *IEEE TRANS. PAMI, 22(8):747-757, August 2000.*

[15] *K. Rohr,* Towards Model-based Recognition of Human Movements in Image Sequences, *Computer Vision, Graphics, and Image Processing: Image Understanding 59:1 (1994) 94-115.*

[16] *Y. Yacoob, M.J. Black,* Parameterized Modeling and Recognition of Activities, *Journal of Computer Vision and Image Understanding, 73(2), 1999, 232-247.*