# TASK INFERENCE USING EYE GAZE TRACKS AND HIDDEN MARKOV MODELS


P.Yaswanth Kumar (10472)

Jitendra Kumar Dhakar (10316)

Mentor: Prof. Amitabha Mukerjee

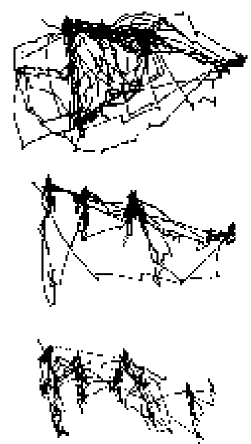
Dept. Of Computer Science and Engineering

# ABSTRACT

From the past work, it is shown that visual tasks such as counting, searching, observing highly influence the eye movement patterns of the observer and so the resulting eye gaze trajectory obtained for each task. Yarbus process shows that different eye gaze trajectories are obtained based on different visual tasks the viewers are given. Our main aim is to implement both Yarbus process and Inverse Yarbus Process whereby we can infer the visual task by observing the measurements of a viewer's eye movements while executing the visual task. The method we are using is using Hidden Markov Models (HMMs) to infer the viewer's task from the eye movements of the viewer.

# RELEVANT WORK DONE

## Yarbus Process :



[2] [2]

This shows that different eye gaze trajectories are obtained for different tasks and center of gaze mainly depends on the fixation points.

Tasks given to the viewer in the above picture are

- Free examination of the picture
- Estimate ages of the people in the picture.
- Remember their clothes.

# INTRODUCTION

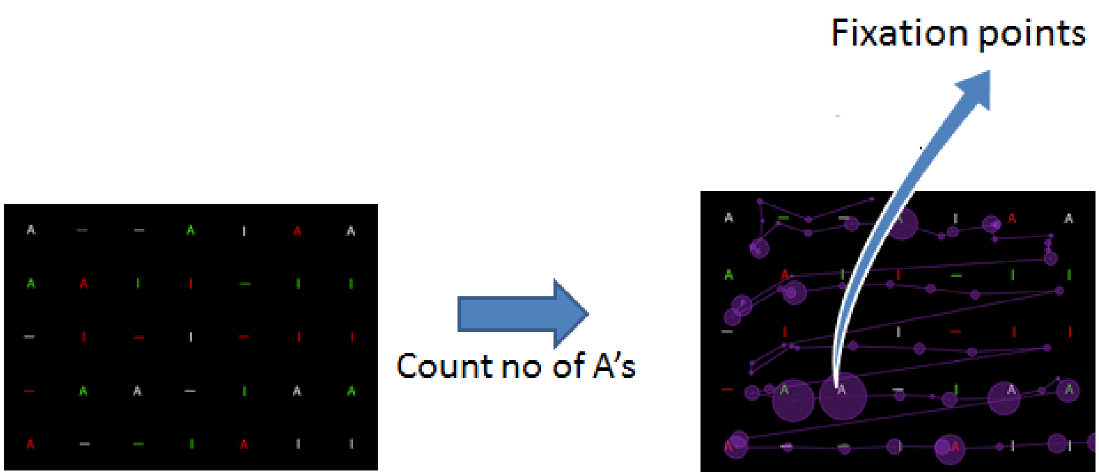
## Inverse Yarbus Proccess :


In Inverse Yarbus Process, we are given an eye gaze trajectory of an image as an input and we have to find the **TASK** from which it might had come.

Many methods exist for Yarbus process and very less for Inverse Yarbus process. Our aim is to implement both Yarbus and Inverse Yarbus Process using Hidden Markov Models(HMMs).

# METHODOLOGY

## Collecting Data :



This is our input image.
We have 4 tasks to train on these images. They are :

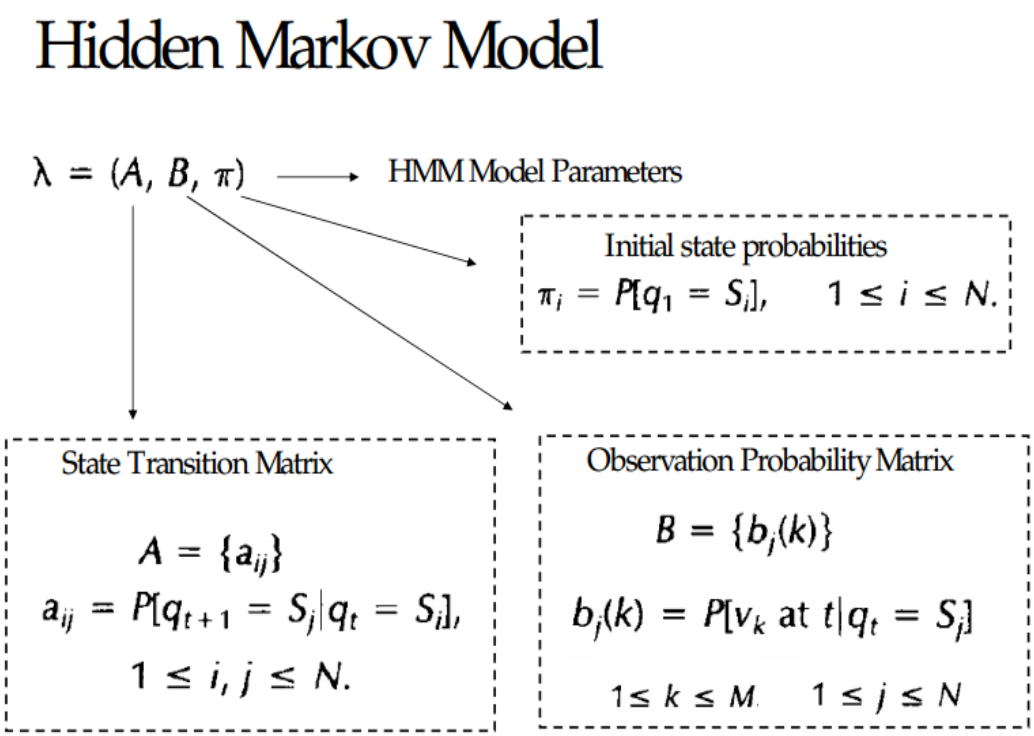
Counting number of

(1) Character A’s
(2) Green bars
(3) Horizontal bars
(4) Vertical bars

For each task

- Get the people
- Give them the task
- Collect eye gaze trajectories obtained

This way we collect all the eye gaze trajectories obtained for each task. Some of them will be used for training and some for testing.




[3]

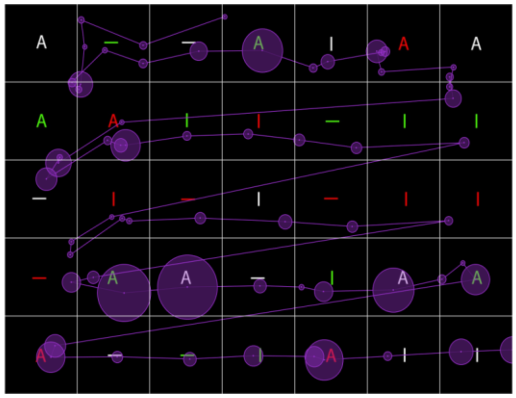
A := State Transition Matrix = Probability to go from one hidden state ‘i’ to another hidden state ‘j’

B := Observation Probability Matrix = Probability that a symbol ‘k’ is emitted from the hidden state ‘j’

π := Initial State Probabilities = Probability that state ‘i’ is the starting state of the model.

# TRAINING HMMs

Now we train a HMM (Hidden Markov Model) for each task.



Assume the eye gaze trajectory to be a 7*5 grid. Collect top 20 to 23 points from this image based on their intensity, find their co-ordinates and assign them a state based on which box they are present in. This is called Observation sequence.

For example, the Observation sequence obtained from the above image is
9,3,4,5,6,14,15,15,9,18,22,23,24,25,26,27,28,29,29,31,32,33,33,35,35

In this way we collect this set of observation sequences for each image for a task and send that matrix as an input to the BAUM-WELACH Algorithm.

## BAUM-WELCH ALGORITHM :

Constructs a HMM for each task by taking the observed sequence of states matrix obtained.

$$\lambda = (A, B, \pi)$$

$$[\ \pi,\ A,\ B\ ] = \text{dhmm\_em}(\text{data},\ \pi_e,\ A_e,\ B_e\ ,\text{'max\_iter'},5);$$

$\pi_e$, $A_e$, $B_e$ are the guess (random) probability matrices corresponding to the parameters of the HMM respectively.

In this way, we train each HMM (totally 4 HMMs)

# TASK INFERENCE (TESTING)

Now we take an eye gaze trajectory image and find its observation sequence similarly and input this sequence to FORWARD Algorithm.

Loglik = dhmm_logprob(data_new, π, A, B);

Forward algorithm calculates the loglikelihood value for each HMM and the task with the maximum value of the likelihood value is the REQUIRED Task.

# RESULTS

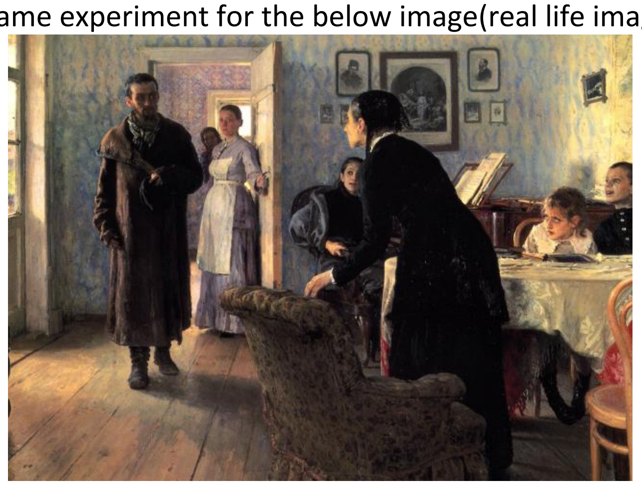
RESULT <4x8 double>

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | -68.1936 | -60.4541 | -80.6011 | -84.8511 | -78.3745 | -75.6521 | -82.2972 | -69.2814 |
| 2 | -80.0594 | -72.5376 | -55.6824 | -66.2969 | -73.5374 | -72.4116 | -69.0423 | -66.4744 |
| 3 | -73.7965 | -76.4363 | -65.8082 | -75.5385 | -70.7605 | -68.5534 | -73.4736 | -70.9167 |
| 4 | -75.4621 | -72.2702 | -67.9874 | -63.6914 | -73.4466 | -77.8487 | -61.2939 | -66.0169 |

We trained for 8 images (2 for each task). We can see that in 7 out of 8 cases, the inferred task is correct. The wrong one was for $2^{nd}$ task of counting number of green bars.

## RESULTS FOR NEW DATA :

We also did the same experiment for the below image(real life image).

[1] 

The tasks we considered are :

(1) Estimate ages of the people
(2) Guess the last time the man in the picture visited the house
(3) Estimate the wealth of the family (how rich or poor they are)
(4) Observe and remember clothes of the people

The Results obtained are:

RESULT <4x16 double>

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -37.83... | -19.8590 | -41.0743 | -31.13... | -43.60... | -44.48... | -38.74... | -41.6247 | -43.30... | -38.9878 | -63.83... | -46.6666 | -58.36... | -64.6511 | -59.05... | -58.3633 |
| 2 | -45.91... | -23.2596 | -38.3275 | -32.85... | -47.46... | -47.74... | -37.25... | -61.0985 | -91.27... | -35.7233 | -73.56... | -93.2372 | -57.42... | -86.8224 | -81.72... | -75.9651 |
| 3 | -40.06... | -86.0354 | -39.8681 | -38.51... | -44.13... | -49.39... | -33.84... | -77.2804 | -73.95... | -78.6600 | -78.49... | -67.1412 | -63.59... | -89.3575 | -67.38... | -79.5926 |
| 4 | -38.42... | -34.9283 | -36.5693 | -37.38... | -31.48... | -38.84... | -37.69... | -66.3686 | -50.34... | -45.1400 | -47.55... | -40.9351 | -56.89... | -51.8343 | -56.96.. | -71.9187 |

We can see that for the test cases of task1 and task4, the inferred tasks are pretty good and for task3 and task4, the results are bad. This is because :

- For task1 and task4 the observed eye gaze trajectories (trained data) were almost similar respectively for each of them because the viewer would in general look at their faces for task1 and their whole clothes(body) for task4. So task1 and task4 are trained with good accuracy. So good results for test cases.

- But for task2 and task3, the tasks are somewhat confusing and so we got random eye gaze trajectories for different persons and many of them matched with almost similar to the task1 and task4. So the HMM could not train efficiently and so even for their test cases, we observe that the inferred task is one among task1 and task4.

# FUTURE WORK

- So the above problem may be tackled with some other approach with good accuracy even for those type of cases.

- For better accuracy, we can also train HMMs by directly taking their co-ordinates

- This can be extended to motion pictures or videos also.

# ACKNOWLEDGEMENT

We thank our advisor Dr.Amitabha Mukerjee for providing such support and references throughout the project. We also thank Prof. Arvind for helping us in getting dataset. We also thank Ankit Gupta for his support in providing useful information for understanding HMMs.

# REFERENCES

[1] **Haji-Abolhassani, A.** and Clark, J.J., "Visual Task Inference Using Hidden Markov Models", proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp. 1678--1683, 2011

[2] A.L. Yarbus. Eye movements during perception of complex objects. Eye movements and vision, 7:171–196, 1967.

[3]http://www.cs.umd.edu/~djacobs/CMSC828/ApplicationsHMMs.pdf

[4] HMM Toolbox : http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html