

ESE 650 Project 3 report: Gesture recognition

Dhruva Kumar

1 Introduction

Given the gyroscope and accelerometer readings of 6 different gestures (circle, eight, inf, etc), the task is to identify the gesture for a new sequence.

Hidden Markov Models are used to learn the hidden states given the observation sequence (gyroscope and accelerometer readings) for each gesture. This is done via the EM like procedure: Baum Welch algorithm. For testing, the forward step is used to find the likelihood of the sequence given each model and the maximum is taken as the most predicted sequence.

2 Data preprocessing

The raw gyroscope and accelerometer readings are used as the observation sequences/features to train the HMM model after processing it:

1. The data (gyroscope and accelerometer readings) are mean centered.
2. The data is then smoothed using the Savitzky Golay filter. This is a FIR filter which maximizes the SNR (signal to noise ratio) based on regression. A low pass filter and an adaptive filter, weiner filter was tried and tested. The Savitzky Golay filter outperformed these two. It smoothed out the signals better and maintained the periodicity among the repeating sequences. The weiner filter did not work so well since because of its adaptive nature, the periodicity among the repeating sequences was lost. This periodicity is important to learn the states for the particular gesture.
3. On observation of the data, it was found to contain static parts in the starting and end. Since this does not contribute to the states, it was removed. The trimming was done by finding when the slope of the signal crossed a certain threshold and that was marked as the start point. The same was done for the end point.

Quantization of observation After cleaning up the data, the sequences were discretized using kmeans clustering. 20 clusters were chosen to split the data into. The six dimensional centroids were retained to quantize new test sequences. The norm of the new test sequence and every centroid was computed and the minimum taken to find the cluster which the new sequence closely belongs to.

3 Initial model parameters

After cross validation, the following initial parameters were chosen for the final model:

Number of states (N): 10

Number of clusters (M): 20

HMM structure: Left to right with the last node connected to the first one and self loops.

π : [1,0,0...N] - begins with the first state.

A: band diagonal (based on the structure) of random values

B: random values

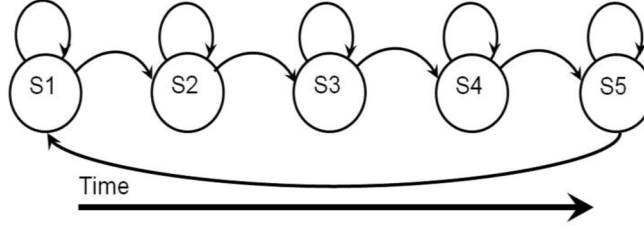


Figure 1: HMM left to right structure used

A left to right model worked slightly better than a fully connected one for different values of N and M . After convergence of the learning procedure, it was found that the transition matrix A of the fully connected structure more or less resulted in a diagonal matrix, implying an inherent left to right structure in the model. A left to right model converged faster and sometimes gave better results.

Vectorization: To speed up the process of cross validation over different parameters, the code was substantially vectorized, specifically at the forward, backward and the update of A and B steps.

Table 1: Effect of vectorization for 5 sequences and 1 iteration

Vectorization	Time taken (seconds)
Unvectorized code	4.514
Vectorized	0.86

4 Learning: Baum Welch

The HMM model was learnt using the Baum Welch procedure. It is an iterative EM procedure alternating between the E and M step, trying to maximize the cost function $P(O|\lambda)$ until a convergence criteria is met.

Initialization:

The initial model $\{\pi, A, B\}$ is taken as described in the section above.

Expectation: Given model, compute the forward and backward variables α and β

This comprises of the forward and backward step. It computes $P(O|\lambda)$, the probability of the likelihood of the observation sequence, given the model by computing the forward α and backward β variables.

The scaling as described in the Rabiner paper [1] is used.

- Forward procedure (compute α):

1. Initialization:

$$\begin{aligned}
 \alpha_1(i) &= \pi_i b_i(O_1) & 1 \leq i \leq N \\
 c_1 &= \frac{1}{\sum_{i=1}^N \alpha_1(i)} \\
 \hat{\alpha}_1(i) &= c_1 \alpha_1(i)
 \end{aligned}$$

2. Induction:

$$\begin{aligned}\alpha_{t+1}(j) &= \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) & 1 \leq t \leq T-1, 1 \leq j \leq N \\ c_t &= \frac{1}{\sum_{i=1}^N \alpha_t(i)} \\ \hat{\alpha}_t(i) &= c_t \alpha_t(i)\end{aligned}$$

3. Termination:

$$\begin{aligned}P(O|\lambda) &= \sum_{i=1}^N \alpha_T(i) \\ &= \frac{1}{\prod_{\tau=1}^T c_\tau} \\ &= \frac{1}{C_T} \\ \log P(O|\lambda) &= - \sum_{t=1}^T \log c_t\end{aligned}$$

- Backward procedure (compute β):
(Using the same scaling computed from α)

1. Initialization:

$$\begin{aligned}\beta_T(i) &= 1 & 1 \leq i \leq N \\ \hat{\beta}_T(i) &= c_T \beta_T(i)\end{aligned}$$

2. Induction:

$$\begin{aligned}\beta_T(i) &= \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) & T-1 \geq t \geq 1, 1 \leq i \leq N \\ \hat{\beta}_t(i) &= c_t \beta_t(i)\end{aligned}$$

For multiple sequences, α^l , β^l and c^l for each sequence l is computed.

Modification: Given α and β , compute most likely model $\{\pi, A, B\}$.

We adjust $\{\pi, A, B\}$ to maximize the cost function: $P(O|\lambda)$. The re-estimated parameters $\bar{\pi}, \bar{A}, \bar{B}$ are as

follows:

$$\begin{aligned}
\bar{\pi}_i &= \text{expected number of times in state } S_i \text{ at time } (t = 1) \\
&= \gamma_1(i) \\
\bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to } S_j}{\text{expected number of transitions from state } S_i} \\
&= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\
&= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(j)} \\
&= \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \hat{\beta}_t(j) / c_t}
\end{aligned}$$

For multiple observations,

$$= \frac{\sum_{l=1}^{L-1} \sum_{t=1}^{T_l-1} \hat{\alpha}_t^l(i) a_{ij} b_j(O_{t+1}^l) \hat{\beta}_{t+1}^l(j)}{\sum_{t=1}^{T_l-1} \hat{\alpha}_t^l(i) \hat{\beta}_t^l(j) / c_t^l}$$

$$\begin{aligned}
\bar{b}_j(k) &= \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} \\
&= \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \\
&= \frac{\sum_{t=1, O_t=v_k}^T \hat{\alpha}_t(j) \hat{\beta}_t(j) / c_t}{\sum_{t=1}^T \hat{\alpha}_t(j) \hat{\beta}_t(j) / c_t}
\end{aligned}$$

For multiple observations,

$$= \frac{\sum_{l=1}^L \sum_{t=1, O_t^l=v_k}^{T_l} \hat{\alpha}_t^l(j) \hat{\beta}_t^l(j) / c_t^l}{\sum_{t=1}^{T_l} \hat{\alpha}_t^l(j) \hat{\beta}_t^l(j) / c_t^l}$$

$\gamma_t(i)$ is the probability of being in state i at time t given the observations sequence and the model λ

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{P(O | \lambda)}$$

$\xi_t(i, j)$ is the probability of being in state S_i at time t and S_j at time $t+1$, given the model and observation sequence.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)}$$

Convergence criteria: The convergence criteria was assigned as when the absolute of the difference in the log likelihood of $P(O | \lambda)$ between successive iterations is less than 0.1

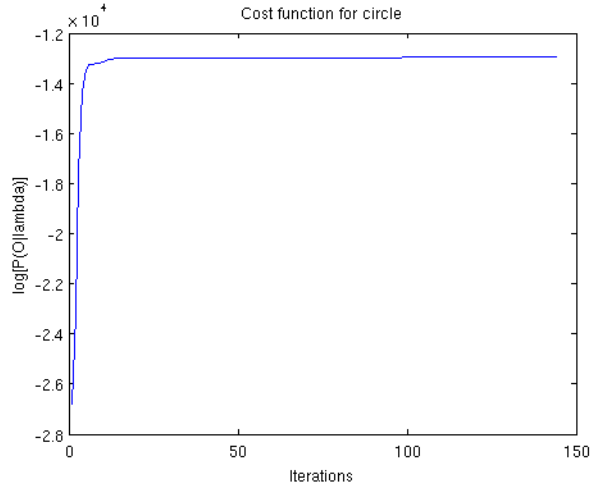


Figure 2: Cost function for circle after running Baum Welch

5 Testing and results

Once the models were generated for each gesture, testing on the test sequences was done using the forward procedure. The forward procedure gives us $P(O|\lambda)$, i.e. the likelihood of the test sequence, given the models. The maximum value among the models was chosen as the most likely gesture.

Table 2: Confusion matrix for the in sample training data is given below

	beat3	beat4	circle	eight	inf	wave
beat3	4	1	0	0	0	0
beat4	1	4	0	0	0	0
circle	0	0	5	0	0	0
eight	0	0	0	5	0	0
inf	0	0	0	1	4	0
wave	0	0	0	0	0	5

The accuracy on the **sample test** set given = **100%**

Cross validation was done by setting aside 10 random sequences from 30 sequences of the training data as test sequences. The rest were used for training.

Table 3: Cross validation accuracy for different parameters

N	M	CV accuracy
10	20	80%
15	20	70%
15	15	70%
25	20	80%
35	20	60%
45	20	60%
50	20	70%
70	20	60%

Table 4: Cross validation accuracy with N=10, M=20 for different structures

HMM structure	CV accuracy
Left to right with self loops and last node connected to the first one	80%
Fully connected	80%

The HMM structure did not make much of a difference. The fully connected model reduced somewhat to a left-right model after learning. The left-right model was much faster and therefore used.

References

- [1] Lawrence R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol. 77, No. 2, pp.257-286, February, 1989
- [2] <http://courses.media.mit.edu/2010fall/mas622j/ProblemSets/ps4/tutorial.pdf>