IIT Kanpur – CS 772 : Image Classification Using HMM

Abhinav Jain, Peshal Agarwal, Rishab Jain

Mentored by - **Piyush Rai**

Abstract

This project aims at the development of a man-machine interface using a camera to interpret gestures. Gesture recognition can be conducted with techniques from computer vision, image processing and machine learning. Many approaches have already been made and a lot of algorithms are available for particular tasks in hand.

In this project we are surveying different algorithms like HMM, SVM for image classification.

Here, we have used Hidden Markov Model with Gaussian outputs for 3-Class Image Classification.

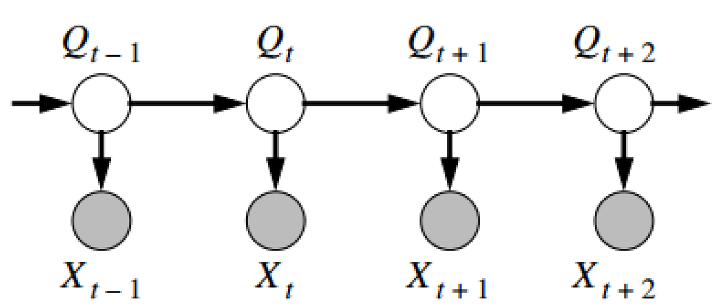


Figure 1: Hidden Markov Model following 1st Order Markov Chain

Objectives

We are training 3 HMM models for classification of 3 kinds of gestures:

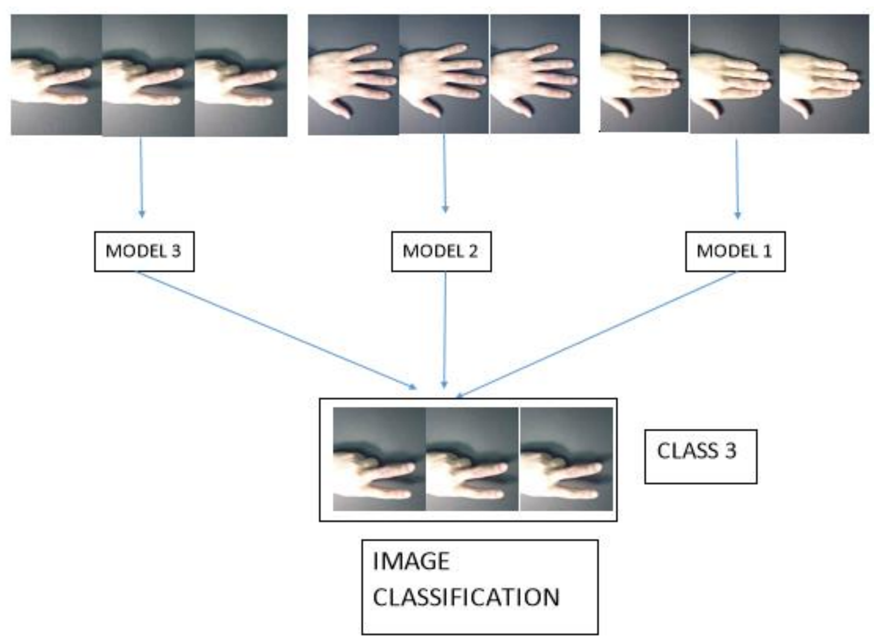


Figure 2: Model Training and Testing on Sequential Data

Feature Vector Representation

The feature vector consists of SIFT features computed on a regular grid across the image (`dense SIFT') and vector quantized into visual words. The frequency of each visual word is then recorded in a histogram for each tile of a spatial tiling as shown. The final feature vector for the image is a concatenation of these histograms. This process is summarized in the figure below:

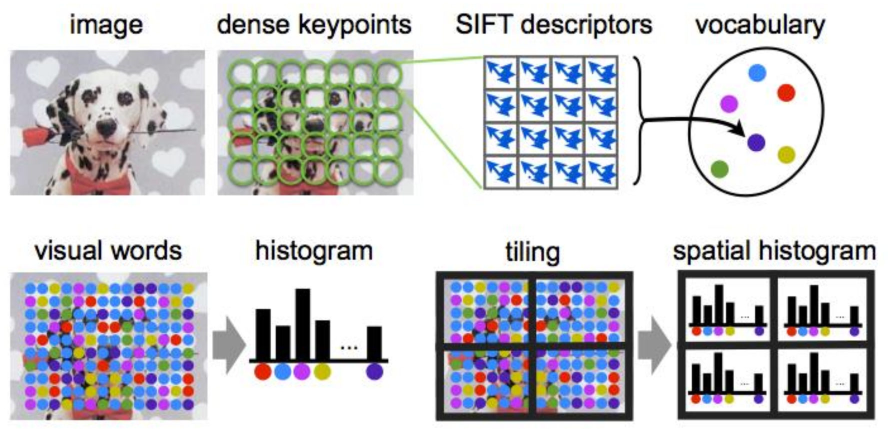


Figure 3: Computing Vocabulary and extracting feature vectors from the image

Hidden Markov Model

Notations:

Observation sequence : $O = O_1 O_2 \cdots O_T$

State sequence : $Q = q_1 q_2 \cdots q_T$

While implementing Hidden Markov Model one faces the following problems-

Problems:

1. Deciding what the states in the model correspond to, and then deciding how many states should be in the model. i.e. choose the model which best matches with the observations.

2. How do we adjust the model parameters .

Solution:

We use the forward-backward procedure to solve the 1st problem, which makes the computation much simpler.

Given the parameters, we calculate the probability of an observation sequence

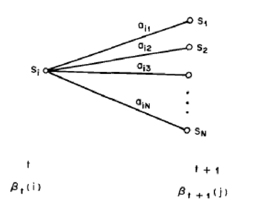
Assumptions: Observations are statistically independent.

1st order markov Chain used.

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda)$$

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T|q_t = S_i, \lambda)$$

Forward Step



1) Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N.$$

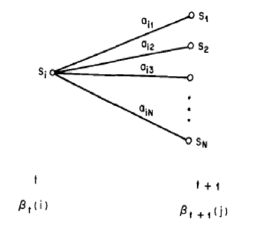
2) Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad 1 \le t \le T-1, \quad 1 \le j \le N.$$

3) Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

Backward Step



1) Initialization:

$$\beta_T(i) = 1, \quad 1 \le i \le N.$$

2) Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1, \quad 1 \le i \le N.$$

Solution to problem 2: there is no optimal way of estimating the model parameters. We can, however, choose $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximized using an iterative procedure such as the Baum-Welch method (or equivalently the EM (expectation-modification)).

Define: $\gamma_t(i) = P(q_t = S_i|O, \lambda)$ $\quad \gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\,\beta_t(i)}$

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j|O, \lambda).$$

$$\xi_t(i, j) = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j).$$

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} = \frac{\sum_{t=1,\ \text{s.t. } O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

Problems Encountered

1. HMM assumes Discrete alphabets as Inputs. But The input data is a continuous feature vector instead of a discrete alphabet. We can **resolve** this using:

- **Vector quantisation**: Each continuous observation vector is mapped into a discrete codebook index. VQ partitions the training vectors into M (size of the codebook) disjoint sets, which are represented by a single vector. These representative vectors can be obtained using K-means clustering.
- **HMM with Gaussian outputs :**

$$b_j(\mathbf{O}) = \sum_{m=1}^{M} c_{jm} \mathfrak{N}[\mathbf{O}, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm}], \quad 1 \le j \le N$$

$$\bar{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j, k) \cdot \mathbf{O}_t}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

$$\bar{\mathbf{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j, k) \cdot (\mathbf{O}_t - \boldsymbol{\mu}_{jk})(\mathbf{O}_t - \boldsymbol{\mu}_{jk})'}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

$$\gamma_t(j, k) = \left[\frac{\alpha_t(j)\,\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)}\right]\left[\frac{c_{jk} \mathfrak{N}(\mathbf{O}_t, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})}{\sum_{m=1}^{M} c_{jm} \mathfrak{N}(\mathbf{O}_t, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})}\right].$$

2. **Parameter Initialization.**

The choices for the number of hidden states and the initial parameter values are crucial to the success of any application of the HMM framework.

- **Number of hidden states**: For the number of hidden states, we tried using values ranging from 2 up to about 8. It should be noted that adding states *always* improves the achievable likelihood. After this initial exploration, all subsequent models were created with five hidden states.
- **Π and mixture weights** : Uniform Initialization
- **Transition Matrix** : We try to heavily bias the model toward remaining in the same state from one time step to the next. Hence, the diagonal values of the transition matrix were initialized to values distributed around 0.99.

Dest. State (columns); Start State (rows)

.977		.001	.021	
	.997			.003
.005		.992	.003	.0001
.008		.004	.982	.006
	.012		.0006	.988

Figure 4: Prior on transition matrix

- **Observation (emission) matrix:** It had little effect on the HMMs produced. So, we initialized it with uniformly distributed values.
- In case of Gaussian HMM we have two other parameters namely, mean vector and the covariance matrix.

Initial estimates for parameters of mixture of Gaussians were done using 'k-means' or by choosing centres randomly from data and using covariance between data for computing covariance matrix (which can be diagonal, spherical or full matrix).

3. In Case of limited Training Data:

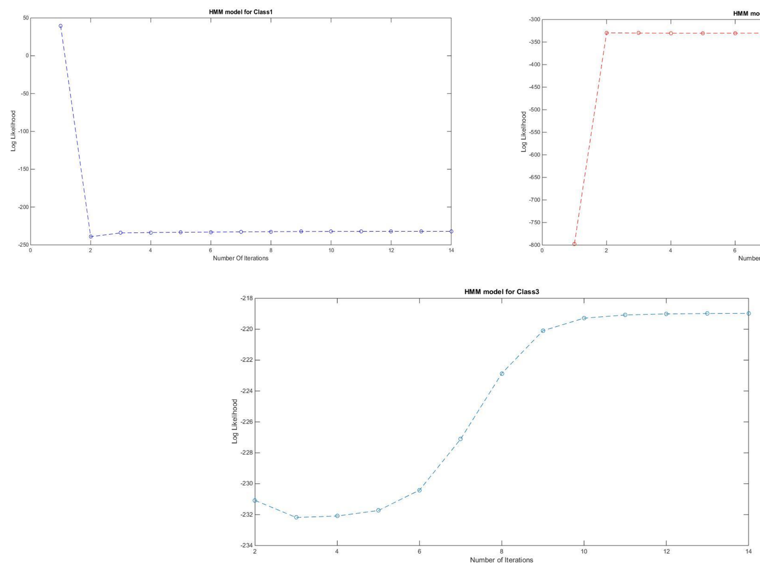
For the continuous models, we found that it is preferable to use diagonal covariance matrices with several mixtures(i.e. higher M), rather than fewer mixtures with full covariance matrices. The reason for this is simple, namely the difficulty in performing reliable re-estimation of the off diagonal components of the covariance matrix from the necessarily limited training data.

Model extension – 2-Dimensional Hidden Markov Model

Results

HMM model was trained for the classification of 3 kinds of gestures using **batch training**. Each model was trained using the same prior and maximum number of iterations allowed were 20.

As a result log likelihood increased in each iteration with model converging around 15th iteration.



We classified a sequence into one of 3 classed by training up 3 HMMs, one per class and then computed the log-likelihood that each model gave to the test sequence; if the i[th] model was the most likely, then declared the class of the sequence to be class 'i'.

Conclusion

Limitations of HMM-

- A major limitation of using HMM is the assumption that successive observations (frames of speech) are independent, and therefore the probability of a sequence of observations.
- Markov assumption itself, i.e., that the probability of being in a given state at time t only depends on the state at time t - 1, is clearly inappropriate for instances where dependencies often extend through several states.

However, in spite of these limitations this type of statistical model has worked extremely well for certain types of speech recognition and image classification problems.

References

1. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition *LAWRENCE R. RABINER, FELLOW, IEEE.*
2. https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm_usage.html.
3. 1D and Pseudo-2D Hidden Markov Models for Image Analysis. Theoretical Introduction *Stephane Marchand-Maillet* - Multimedia Communications.