Hidden Markov Models (HMM)

Summary

- Motivation
- HMMs
- Estimation of Hidden Variables
- Learning
- Examples

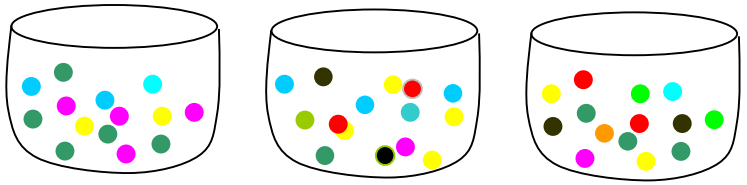
Motivation

Hidden Markov models have been used in several problems:

- speech recognition
- hand written text recognition
- tracking of human gestures
- self localization of mobile robots
- grammatical analysis

All these problems can be tackled by using models based on two random processess: an observable process (visible) and an unobserved process (hidden).

Challenge



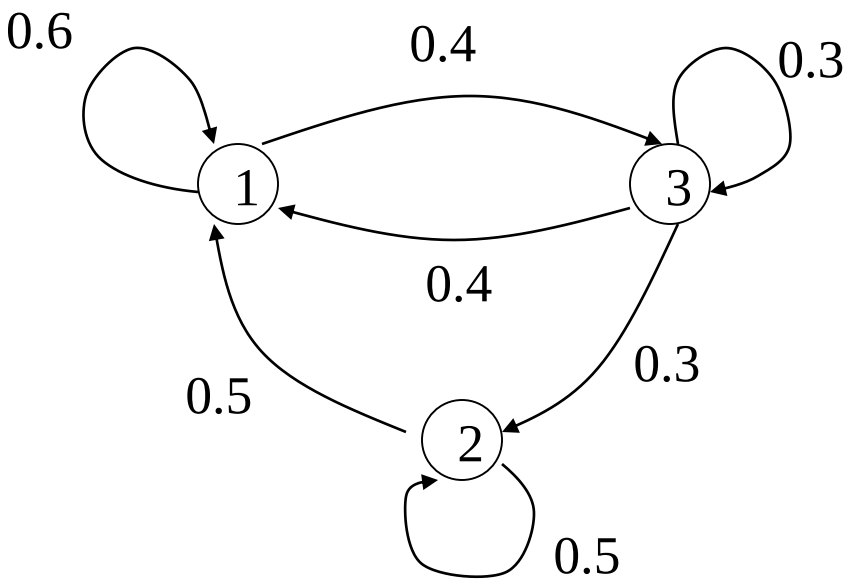
Observations:

What is the sequence of boxes ?

Hypothesis:

- each ball is extracted from one of the boxes.
- boxes are randomly selected and this choice only depends on the last chosen box.

Label Generation



Label sequence: 3 3 2 1 1 3 1 1 3 2 2 2 1

Challenge - Formulation

There are two random processes in the box problem:

- sequence of colored balls $y_1, y_2, \ldots, y_N$ (visible)
- sequence of the box labels $x_1, x_2, \ldots, x_N$ (invisible)

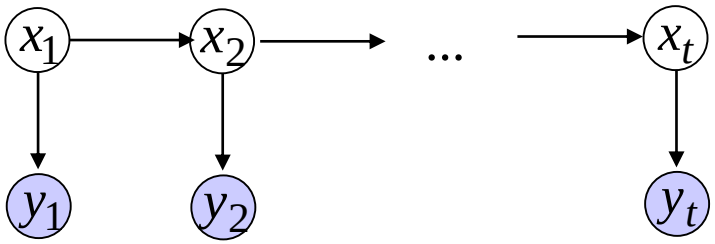
$\{x_t\}$ is a 1st order Markov process i.e., $P(x_t|x_1,\ldots, x_{t-1}) = P(x_t|x_{t-1})$

y_t depends on x_t: $P(y_t|x_t)$

y_t is independent on x_τ, $\forall\ \tau<t$, if x_t is known

This model is known as a ***Hidden Markov Model.***

HMM



$x_1,\ldots,x_t$ sequence of state variables

$y_1,\ldots,y_t$ sequence of observations

The **state variables** are **discrete,** $x_i \in \{1,\ldots,N\}$

and they verify a 1st order Markov property.

Characterization

Sequence of state varibles:

initial distribution $P\{x_1 = i\} = \pi_i$ vector π

transition probabilities: $P\{x_t = j \mid x_{t-1} = i\} = a_{ij}$ matrix A

Sequence of observations: emission probabilities

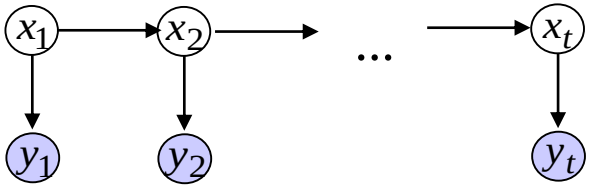
discrete case: $P\{y_t = j \mid x_t = i\} = b_{ij}$ matrix B

Continuous case: $p(y_t \mid x_t = i) = b_i(y_t)$

Example

Model:

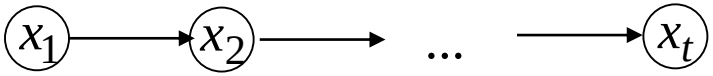
$$\pi = \begin{bmatrix} .2 \\ .8 \end{bmatrix} \quad A = \begin{bmatrix} .9 & .1 \\ .3 & .7 \end{bmatrix} \quad \mathrm{B} = \begin{bmatrix} .2 & .7 & .1 \\ .6 & .1 & .3 \end{bmatrix}$$



Sequences generated by the model

x: 2 2 1 1 1 1 2 2 2 1 1 1 2 1 1 2 2
y: 1 1 2 1 2 2 1 1 3 2 1 2 1 2 2 3 1

Uncertainty Propagation



If no observations are available, the state distribution evolves according to

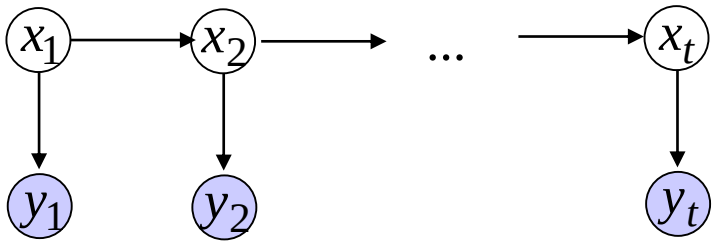
$$\pi^t = A'\pi^{t-1}, \qquad \pi^1 = \pi \qquad \text{prediction}$$

where

$$\pi^t = [\pi_1^t ... \pi_N^t]', \qquad \pi_i^t = P\{x_t = i\}$$

The asymptotic distribution verifies $(A'-I)\pi=0$

Uncertainty Propagation



When there are observations available, $\pi_t = col\{P(x_t = i / y_{1:t})\}$
Is updated according to:

Filtering $\pi^t = kD_{y_t}\tilde{\pi}^t \quad \mathrm{D}_{y_t} = diag(b_1(y_t),...,b_N(y_t))$

Prediction $\tilde{\pi}^{t+1} = A'\pi^t$

These equations correspond to the Kalman filter in the case of continuous state variables. However, they do not require any linearity or Gaussianity assumptions.

Likelihood Function

Joint distribution

$$p(x,y) = \prod_{t=1}^{n} a_{x_{t-1}x_t} p(y_t \mid x_t) \qquad \text{convention} : a_{x_0 x_1} = \pi_{x_1}$$

Log likelihood function

$$p(y) = \sum_{x} \left[\prod_{t=1}^{n} a_{x_{t-1}x_t} p(y_t / x_t) \right]$$

This expression is too complex to be useful!

Forward Backward Algorithms

(Baum)

Recursively computes

$$\alpha_t(i) = P(x_t = i, y_{1:t}), \quad \beta_t(j) = P(y_{t+1:n} \mid x_t = j)$$

Forward Algorithm

$$\alpha_1(j) = b_j(y_1)\pi_j$$

$$\alpha_t(j) = b_j(y_t) \sum_{i=1}^{N} a_{ij}\alpha_{t-1}(i) \quad \text{t} = 2, \ldots, \text{n}$$

Backward Algorithm

$$\beta_N(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1})\beta_{t+1}(j) \quad \text{t} = \text{n} - 1, \ldots, 1$$

Likelihood function

$$P(y) = \sum_{i=1}^{N} \alpha_n(i) \qquad P(y) = \sum_{i=1}^{N} \pi_i b_i(y_1)\beta_1(i)$$

Matrix F-B Algorithm

Forward Algorithm

$$\alpha_1 = D_{y_1}\pi$$

$$\alpha_t = D_{y_1} A'\alpha_{t-1} \quad t = 2, \ldots, n$$

Backward Algorithm

$$\beta_N = 1$$

$$\beta_t(i) = AD_{y_{t+1}}\beta_{t+1} \quad \mathrm{t} = \mathrm{n}-1, \ldots, 1$$

Th forward algorithm is similar to the propagation of the conditional distribution of the state variable, except for a multiplicative factor.

Viterbi Algorithm

It is possible to efficiently calculate the most likely state sequence.
The Viterbi algorithm does this in linear time using Dynamic Programming.

1. Forward recursion

$$\delta_1(i) = \pi_i b_i(y_1)$$

$$\delta_t(j) = b_j(y_t) \max_i \delta_{t-1}(i) a_{ij} \quad t = 2,3,\ldots,n$$

3. Backward recursion

$$\hat{x}_n = \arg\max_i \delta_n(i)$$

$$\hat{x}_t = \arg\max_i \delta_t(i) a_{i\hat{x}_{t+1}} \quad t = n\text{-}1, \ldots, 1$$

Viterbi Training Algorithm

Model training consists of the estimation of π, A, B from known observation sequences (training data). The state sequences x are unknown.

Viterbi training algorithm is iterative. It starts from a set of initial estimates of the unknown parameters and recursively updates them in two steps:

1- estimation of the most probable state sequences for the hidden variables.

2- estimation of matrices π, A, B from the relative frequencies of the state transitions or output symbol emission.

When the observations are continuous (e.g., mistures of Gaussians) step 2 is performed by using standard estimation methods (e.g. EM method).

Viterbi training algorithm is fast but it only considers the most probable sequence.

Baum Training Method

Baum proposed in 1972 training algorithm which takes into account all possible state sequences. Baum algorithm is equivalente to the EM method.

E step:

$$c_i = \sum_{y^p} \frac{\alpha_1(i)\beta_1(i)}{p(y^p)} \qquad d_{ij} = \sum_{y^p} \frac{\sum_{t>1}\alpha_{t-1}(i)a_{ij}b_j(y_t^p)\beta_t(j)}{p(y^i)}$$

$$e_{ij} = \sum_{y^p} \frac{\sum_{t\geq 1: y_t^p = j} \alpha_t(i)\beta_t(i)}{p(y^i)}$$

M step: $\pi_i = \alpha\, c_i\,,\ a_{ij} = \beta\, d_{ij}\,,\quad a_{ij} = \gamma\, e_{ij}$ α,β,γ constants

Training set: $\{y^p\}$

Exercises

1. Given a HMM model $\pi = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ $A = \begin{bmatrix} .3 & .6 & .1 \\ .2 & .5 & .3 \\ .1 & .8 & .1 \end{bmatrix}$ $B = \begin{bmatrix} .8 & .2 & 0 \\ .2 & .7 & .1 \\ 0 & .8 & .2 \end{bmatrix}$

a) compute the asymptotic tate distribution.
b) fiven the sequence y=132 compute the probability distribution of the hidden variables.
c) determine the most likely state sequence using the Viterbi algorithm. Compare with the output of last alina.

2. Consider the HMM defined by:

$\pi = \begin{bmatrix} .7 \\ .3 \end{bmatrix}$ $A = \begin{bmatrix} .7 & .3 \\ .2 & .8 \end{bmatrix}$ $p(y/1) = N(0,1)$ $p(y/2) = .3\,N(-5,1) + .7\,N(5,1)$

After observing y=(.27 –1.43 3.21), compute
a) The distribution of the hidden variables and
b) Most probable state sequence using the Viterbi algorithm.

3. Compute the distribution of the duration time in state i. Discuss the modeling restrictions.

Bibliography

J. Marques, *Reconhecimento de Padrões. Métodos Estatísticos e Neuronais*, IST Press, 1999

L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. IEEE, 257-285, Fev., 1989.