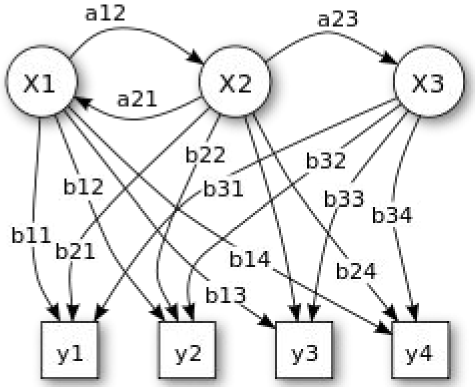
Hidden Markov Models

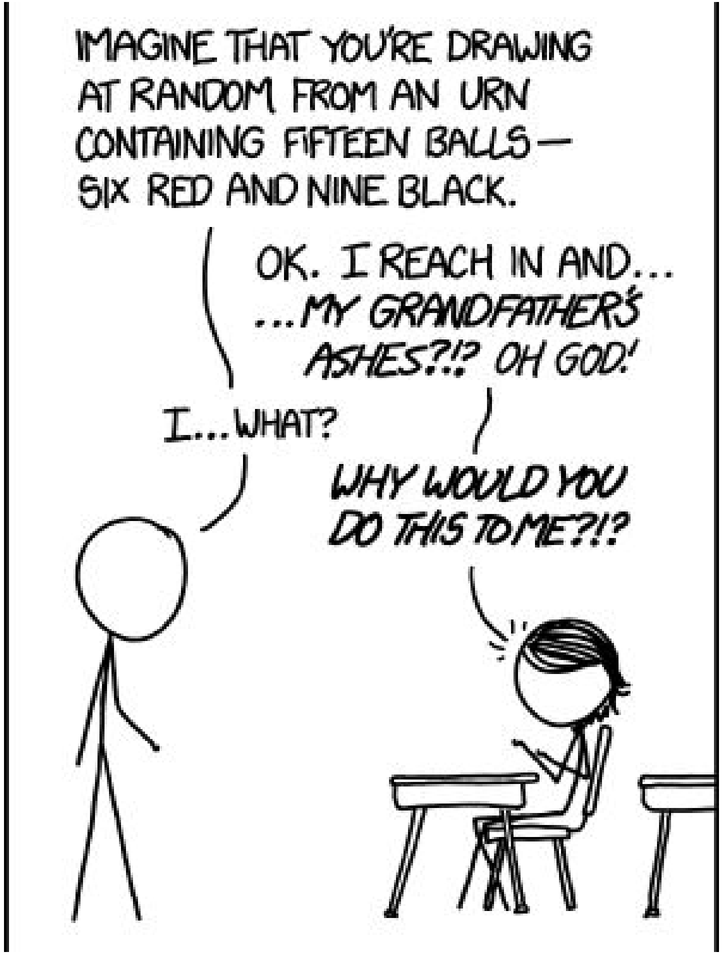
Pratik Lahiri

Introduction

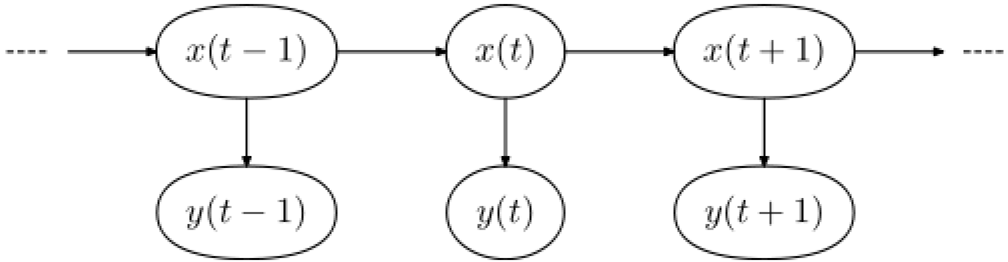
- A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.
- We call the observed event a `symbol' and the invisible factor underlying the observation a `state'.
- An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observable symbols.
- The hidden states form a *Markov chain*, and the probability distribution of the observed symbol depends on the underlying state.
- A generalisation of the Urn problem with replacement.

The Urn Problem





Architecture of HMM



Formal Description of an HMM

O = $\{O1O2,...,ON\}$ Set of possible observations

S = $\{1,2,...,M\}$ Set of possible states

t(i,j) Transition prob

e(x|i) Emission prob

$\pi(i) = \boldsymbol{P}\{y1 = i\}$ for all $i \in S$ Initial state prob

3 Algorithms

- Scoring
- Optimal sequence of states
- Training

Scoring

x = *x*1*x*2 ... *xL* is the observed sequence of length L

So, **y** = *y*1*y*2 ... *yL* is the underlying state sequence

P{x, y | Θ} = P{x | y,}P{y | Θ}, where

P{x | y,} = e(x1 | y1)e(x2 | y2)e(x3 | y3)...e(xL | yL) and

P{y |Θ } = π(y1)t(y1, y2)t(y2 , y3)...t(yL1, yL)

Underlying state is not visible !!

One way to the score is- P{x |Θ } =$\sum_y$ P{x,y | Θ}. (Computationally expensive !!! M^L)

Scoring Contd.

Dynamic Programming- Forward algorithm.

- Forward variable- α(n,i)= P{x1...xn, yn=i|$\mathbf{\Theta}$}
- Recursively, α(n,i)= $\sum_k$ [α(n-1,k)t(k,i)e(x_n | i)]
- P{x | $\mathbf{\Theta}$} = $\sum_k \alpha$(L, k)
- Linear !! O(LM^2)

Viterbi Algorithm (Optimal alignment)

Formally, we want to find the optimal path y^* that satisfies the following-

$y^*=\text{argmax}_y P(y|x,\Theta)$ which is the same as finding the state sequence that maximizes $P\{x,y|\Theta\}$.

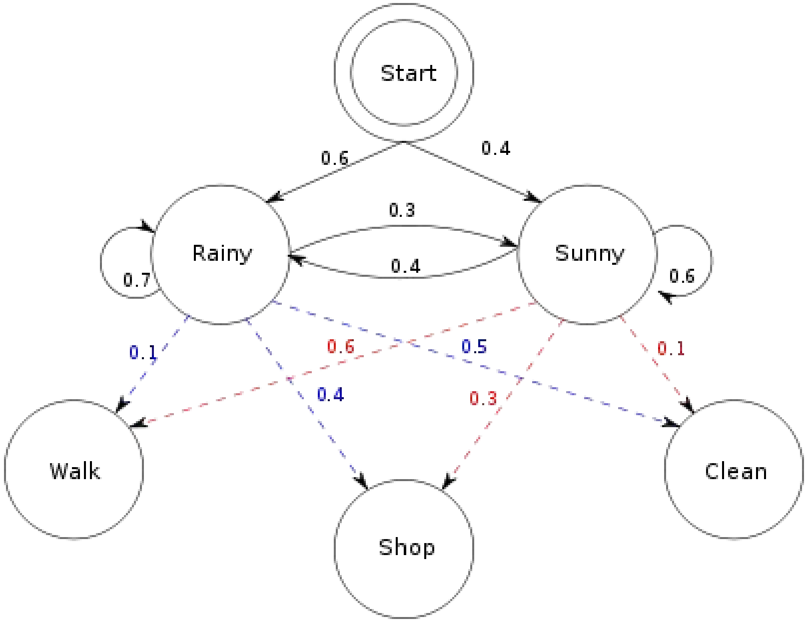
$\gamma(n,i)=\max_{y1..yn-1} P\{x1...xn,y1..yn-1yn=i|\Theta\}$

$\gamma(n,i)=\max_k [\gamma(n-1,k)t(k,i)e(x_n|i)]$

Max prob $P^*=\max_k \gamma(L,k)$

The optimal path y^* can be easily found by tracing back the recursions that led to the maximum probability

Example: Rainy Sunny



Training- Baum Welch

Forward Backward algorithm:

Backward variable: $\beta(n,i)=P\{x_{n+1}...x_L|y_n=i,\boldsymbol{\Theta}\}$

Recursively, $\beta(n,i)=\sum_k [t(i,k)e(x_{n+1}|k)\beta(n+1,k)]$

$\xi_{ij}(n)=P(y_n=i, y_{n+1}=j|x_1..x_L,\boldsymbol{\Theta})=P(y_n=i,y_{n+1}=j,x_1...x_L|\boldsymbol{\Theta})/P(x_1...x_L|\boldsymbol{\Theta})=$

$\alpha(n,i)t(i,j)\beta(n+1,j)e(x_{n+1}|j)$ / $(\sum_i\sum_j\alpha(n,i)t(i,j)\beta(n+1,j)e(x_{n+1}|j))$

$\delta(n,i)=P(y_n=i \mid x_1...x_L,\boldsymbol{\Theta})=\alpha(n,i)\beta(n,i)$ / $\sum_j\alpha(n,j)\beta(n,j)$

Training Contd

Using $\xi_{ij}(n)$ and $\delta(n,i)$ we can estimate the parameters

$\pi(i)=\delta(1,i)$

$t(i,j)=\sum_n \xi_{ij}(n) / \sum_n \delta(n,i)$

$e(x'|y_n=i)= \sum_n 1_{xn=x'} \delta(n,i) / \sum_n \delta(n,i)$

Thanks