Hidden Markov Models

Leonid Zhukov

School of Applied Mathematics and Information Science
National Research University Higher School of Economics

14.11.2013

Hidden Markov Model

HMM formal definition

- Desrete states $S = (S_1 S_2 ... S_N)$
- Observable signals $O = (O_1 O_2 ... O_T)$
- Transion probabilities matrix $A^{N \times N}$, $A_{ij} = P(q_{t+1} = S_j | q_t = S_i)$
- Emmision probabilities matrix $B^{M \times N}$, $B_{ij} = b_i(O_j) = P(O_j | q_t = S_i)$
- Initial states vector π, $\pi_i = P(q_1 = S_i)$
- HMM Model $\lambda = (A, B, \pi)$

Three fundamental problems in HMM

1. The Evaluation problem.
   Given:
   - Observable sequence $O = O_1 O_2 O_3 ... O_T$
   - model $\lambda = (A, B, \pi)$

   Find: $P(O|\lambda)$
2. The Decoding problem.
   Given:
   - Observable sequence $O = O_1 O_2 O_3 ... O_T$
   - model $\lambda = (A, B, \pi)$

   Find: $Q^* = q_1 q_2 q_3 .. q_T = \arg\max_Q P(Q|O, \lambda)$
3. The Learning problem (training).
   Given:
   - Observable sequence $O = O_1 O_2 O_3 ... O_T$

   Find: $\lambda^* = \arg\max_\lambda P(O|\lambda)$

The Evaluation problem

Given:$\lambda = (A, B, \pi)$ and $O = O_1 O_2 O_3 ... O_T$
Find: $P(O|\lambda)$

All possible sequences of states $Q = Q_1 Q_2 Q_3 ... Q_T$

$$P(O|\lambda) = \sum_Q P(O|Q, \lambda) P(Q|\lambda)$$

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T}$$

$$P(O|Q, \lambda) = P(O_1|q_1, \lambda) P(O_2|q_2, \lambda) .. P(O_T|q_T, \lambda) = \\ = b_{q_1}(O_1) b_{q_2}(O_2) ... b_{q_T}(O_T)$$

$$P(O|\lambda) = \sum_{q_1, .. q_T} \pi_{q_1} b_{q_1}(O_1) b_{q_2}(O_2) ... b_{q_T}(O_T) a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T} = \\ = \sum_{q_1, .. q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} ... a_{q_{T-1} q_T} b_{q_T}(O_T)$$

The Evaluation problem

Two MC states example

One time step:

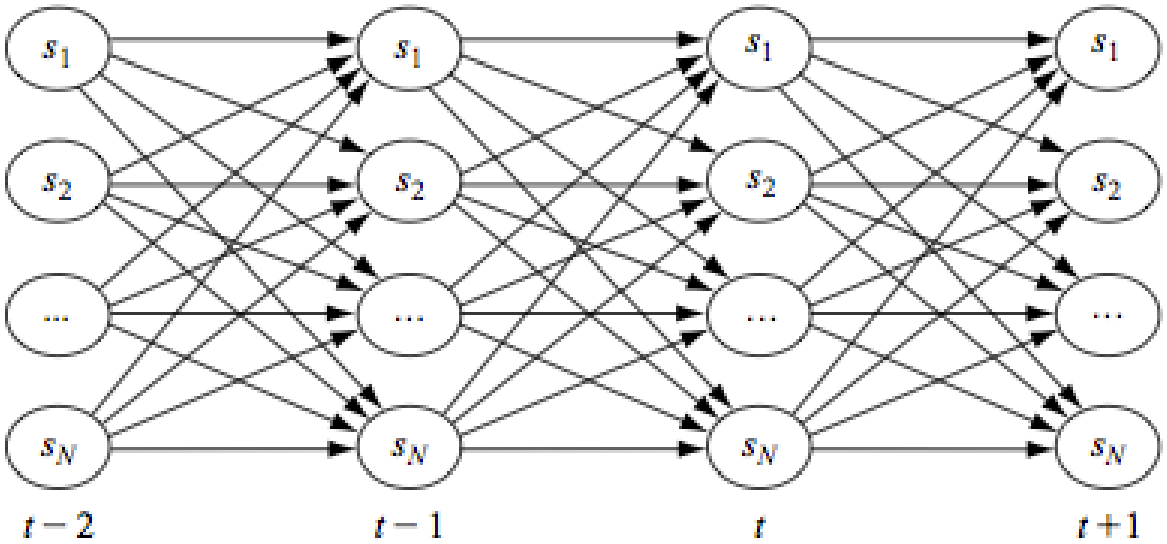
$$P(O_1) = \pi_1 b_1(O_1) + \pi_2 b_2(O_1)$$

Two time steps:

$$\begin{aligned} P(O_1 O_2) = \quad & \pi_1 b_1(O_1) a_{11} b_1(O_2) \quad + \\ & \pi_1 b_1(O_1) a_{12} b_2(O_2) \quad + \\ & \pi_2 b_2(O_1) a_{22} b_2(O_2) \quad + \\ & \pi_2 b_2(O_1) a_{21} b_1(O_2) \end{aligned}$$

Three time steps:

$$\begin{aligned} P(O_1 O_2 O_3) = \quad & \pi_1 b_1(O_1) a_{11} b_1(O_2) a_{11} b_1(O_3) \quad + \\ & \pi_1 b_1(O_1) a_{12} b_2(O_2) a_{22} b_2(O_3) \quad + \ldots \end{aligned}$$



Computational Complexity: $O(2T \cdot N^T)$

Forward algorithm

Partial observation sequence $O_1 \ldots O_t$ that terminates at state S_i
$\alpha_t(i) = P(O_1, O_2 \ldots O_t, q_t = S_i|\lambda),\ t \leq T,\ 1 \leq i \leq N$

then

$$\begin{aligned}
\alpha_1(j) &= P(O_1, q_1 = S_j|\lambda) = \pi_j b_j(O_1) \\
\alpha_2(j) &= P(O_1, O_2, q_2 = S_j|\lambda) = \left[\sum_i \alpha_1(i) a_{ij}\right] b_j(O_2) \\
\alpha_3(j) &= P(O_1, O_2, O_3, q_3 = S_j|\lambda) = \left[\sum_i \alpha_2(i) a_{ij}\right] b_j(O_3) \\
\alpha_{t+1}(j) &= \left[\sum_i \alpha_t(i) a_{ij}\right] b_j(O_{t+1})
\end{aligned}$$

$P(O|\lambda) = \sum_i P(O, q_T = S_i|\lambda) = \sum_i \alpha_T(i)$

Forward-backward algorithm

Forward procedure:

Let $\alpha_t(i) = P(O_1, O_2 \ldots O_t, q_t = S_i|\lambda),\ t \leq T$

1. $\alpha_1(i) = \pi_i b_i(O_1)$
2. $\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_i \alpha_t(i) a_{ij}$
3. $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

Backward procedure:

Let $\beta_t(i) = P(O_{t+1} \ldots O_T|q_t = S_i, \lambda),\ t \leq T$

1. $\beta_T(i) = 1$
2. $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
3. $P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i) \pi_i b_i(0_1)$

Complexity $O(N^2 T)$

Forward-backward algorithm

Algorithm: Forward

Input: $\lambda(A, B, \pi)$, $O = (o_1, ..o_T)$

Output: $P(O|\lambda)$

$\alpha_1(i) = \pi_i b_i(O_1)$

for $j = 1 : N, t = 1 : T - 1$ **do**

$\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_i \alpha_t(i) a_{ij}$

end

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

Forward-backward algorithm

Algorithm: Backward

Input: $\lambda(A, B, \pi)$, $O = (o_1, ..o_T)$

Output: $P(O|\lambda)$

$\beta_T(i) = 1$

for $j = 1 : N, t = T - 1 : 1$ **do**

$\quad \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

end

$P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i) \pi_i b_i(O_1)$

Forward-backward algorithm

forward variable
$\alpha_t(i) = P(O_1, O_2 ... O_t, q_t = S_i|\lambda)$
backward variable
$\beta_t(i) = P(O_{t+1} ... O_T|q_t = S_i, \lambda)$

Then:
$\alpha_t(i)\beta_t(i) = P(O_1, O_2 ... O_t, O_{t+1} ... O_T, q_t = S_i|\lambda) = P(O, q_t = S_i|\lambda)$

and
$P(O|\lambda) = \sum_i P(O, q_t = S_i|\lambda) = \sum_i \alpha_t(i)\beta_t(i)$

Forward-backward algorithm

Algorithm: Forward-backward

Input: $\lambda(A, B, \pi)$, $O = (o_1, ..o_T)$

Output: $P(O|\lambda)$

compute $\alpha_t(i)$

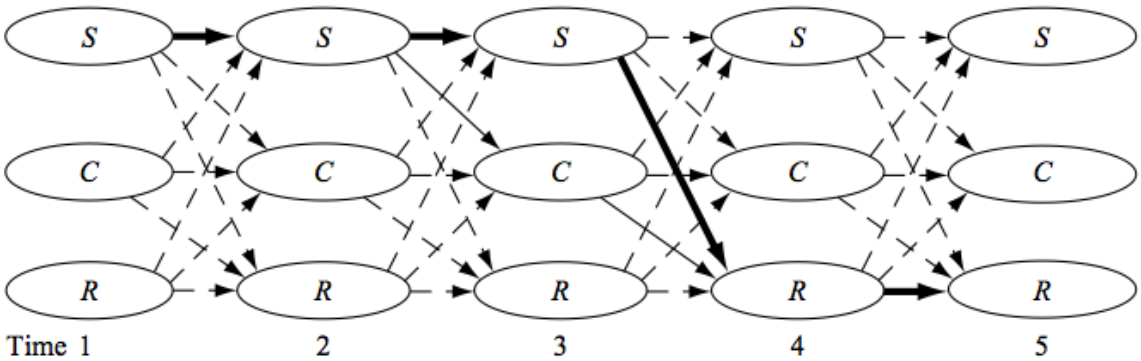
compute $\beta_t(i)$

$P(O|\lambda) = \sum_i \alpha_t(i)\beta_t(i)$

The Decoding problem

Given: $\lambda = (A, B, \pi)$ and $O = O_1 O_2 O_3 ... O_T$
Find: $Q^* = q_1 q_2 q_3 .. q_T = \arg\max_Q P(Q|O, \lambda)$

Select one from all possible sequences of states $Q = Q_1 Q_2 Q_3 ... Q_T$



The Viterbi algorithm

Find the highest probability that partial observation and state sequences up to time t can have, when terminates at S_i.
$\delta_t(i) = max_{q_1,q_2,..q_{t-1}} P(O_1, ..O_{t-1}, q_1, .., q_t = S_i|\lambda)$

$$
\begin{aligned}
\delta_1(j) &= \pi_j b_j(O_1) \\
\delta_2(j) &= \max_{q_1,q_2} P(O_1, O_2, q_2 = S_j|\lambda) = \max_i\{\delta_1(i)a_{ij}\}b_j(O_2) \\
\delta_3(j) &= \max_{q_1,q_2,q_3} P(O_1, O_2, O_3, q_3 = S_j|\lambda) = \max_i\{\delta_2(i)a_{ij}\}b_j(O_3) \\
\delta_{t+1}(j) &= \max_i\{\delta_t(i)a_{ij}\}b_j(O_{t+1})
\end{aligned}
$$

$q^* = \arg\max_j$

The Viterbi algorithm

Algorithm: Viterbi

Input: $\lambda(A, B, \pi)$, $O = (o_1, ..o_T)$

Output: $Q = (q_1 ... q_T)$

$\delta_1(i) = \pi_i b_i(O_1)$

$\psi_1(i) = 0$

for $j = 1 : N, t = 1 : T - 1$ **do**

$\delta_{t+1}(j) = \max_i\{\delta_t(i)a_{ij}\}b_j(O_{t+1})$

$\psi_{t+1}(j) = \arg\max\{\delta_t(i)a_{ij}\}$

end

$q_T^* = \arg\max_i\{\delta_T(i)\}$

for $t = T - 1 : 1$ **do**

$q_t^* = \psi_{t+1}(q_{t+1}^*)$

end

The Learning problem

Given: $O = O_1 O_2 O_3 ... O_T$
Find: $\lambda^* = \arg\max_\lambda P(O|\lambda)$, $\lambda = (A, B, \pi)$

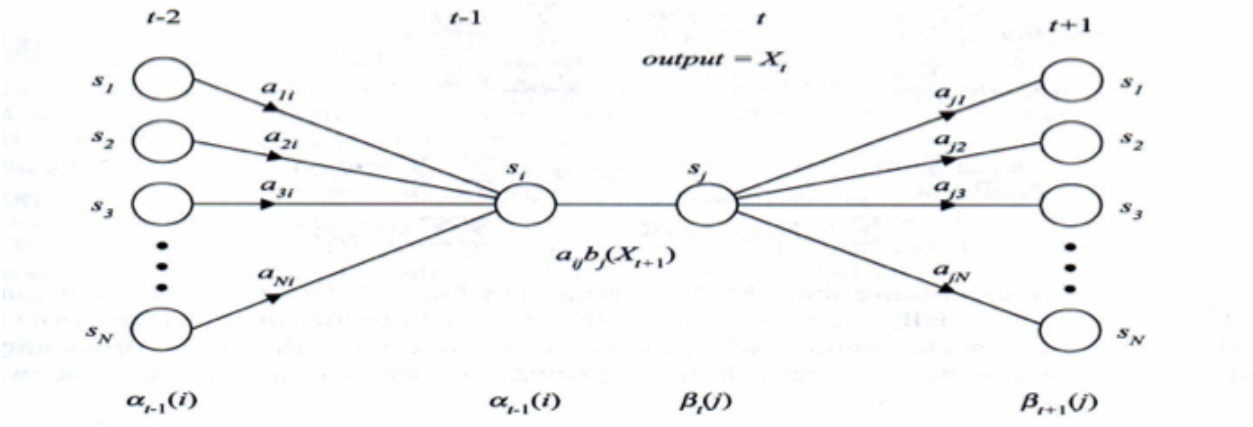
---

forward variable
$\alpha_t(i) = P(O_1, O_2 ... O_t, q_t = S_i|\lambda)$
backward variable
$\beta_t(i) = P(O_{t+1} ... O_T|q_t = S_i, \lambda)$
$P(O, q_t = S_i|\lambda) = \alpha_t(i)\beta_t(i)$
$P(O|\lambda) = \sum_i \alpha_t(i)\beta_t(i)$
$P(O, q_t = S_i|\lambda) = P(q_t = S_i|O, \lambda)P(O|\lambda)$
probability to visit state i at t

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

The Baum-Welch algorithm

transition $i \to j$: probability to visit state i at t and j at $t+1$



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

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

$$[\gamma_t(i) = \sum_j \zeta_t(i,j)]$$

The Baum-Welch algorithm

- $\sum_{t=1}^{T} \gamma_t(i)$ - expected number of times being state i
- $\sum_{t=1,o_t=k}^{T} \gamma_t(i)$ - expected number of times being state i and observing symbol O_k
- $n_i = \sum_{t=1}^{T-1} \gamma_t(i)$ - expected number of transitions from i
- $n_{ij} = \sum_{t=1}^{T-1} \zeta_t(i,j)$ - expected number of transitions from i to j
- Estimations:

$$\bar{\pi}_i = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{n_{ij}}{n_i} = \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1,o_t=k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(j)}$$

- EM type maximization of $P(O|\lambda) = \sum_i \alpha_t(i)\beta_t(i)$

The Baum-Welch Algorithm

Algorithm: Baum-Welch

Input: $O = (o_1, ..o_T)$

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

set intitial random values A, B, π

compute $\alpha_t(i), \beta_t(i), \gamma_t(i), \zeta_t(i, j)$

while $\sum_i \alpha_t(i)\beta_t(i)$ *increasing* **do**

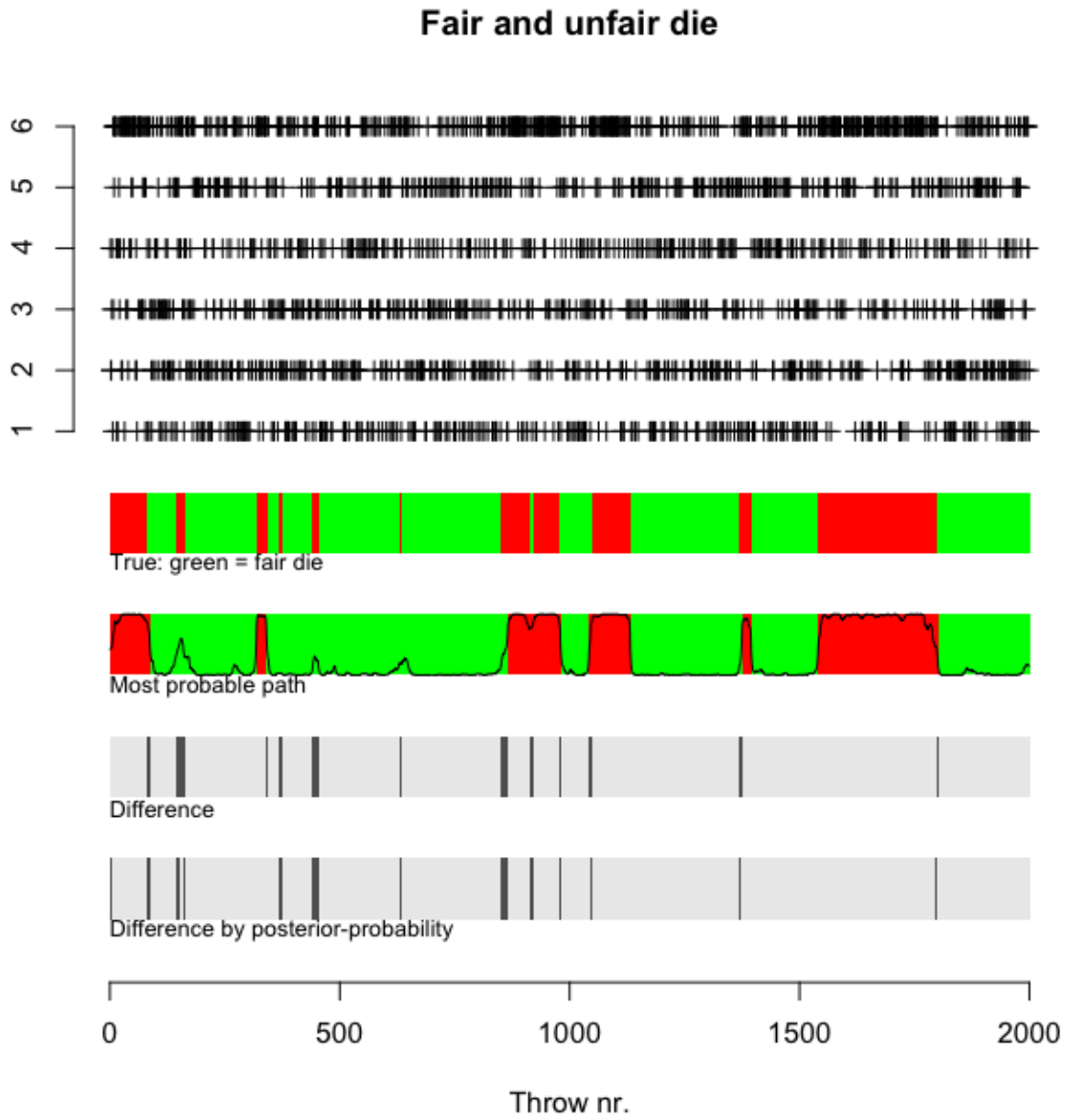
$$\pi_i \leftarrow \gamma_1(i)$$

$$a_{ij} \leftarrow \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j(k) \leftarrow \frac{\sum_{t=1, o_t=k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(j)}$$

end

The dishonest casino



References

- A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Lawrence R. Rabiner. Proc of IEEE, Vol 77, N 2, 1989, pp 257-286