Discrete Markov Process

System is at any time t in one of N states q_t.
At regularly spaced discrete times system undergoes a change of state according to set of probabilities associated with the state.
For the Markov process of first order probability of changing state depends only on current state.

Changing state S_i (at time $t-1$) to state S_j (at time t) is given by:
$P(q_t = S_j \mid q_{t-1} = S_i)$.

Since system state does not depend on time probability of state transition from i to j is:

$$a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i), \quad 1 \le i, j \le N \text{, where } a_{ij} \ge 0 \text{ and } \sum_{j=1}^{N} a_{ij} = 1$$

Observable Markov model – model state corresponds to an observable (physical) event.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Hidden Markov Model

Observable events **are not the states** of the process.
We know at the most the probability of making given observation in every of the process states.
Since we cannot directly observe state changes of the process, such a model is called *Hidden Markov Model* (*HMM*).

A very good introduction to the problem is:
Rabiner L. R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of IEEE, vol. 77, no. 2, February 1989
You can find it easily, for example at:
http://www.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

HMM Parameters

1. The number of states *N* in the model.
2. The number of distinct observations *M* (observation alphabet size)
3. The state transition probability matrix *A*: $A = \{a_{ij}\}$
4. Probability of symbol observation in state *i*:
   $B = \{b_i(k)\}, b_i(k) = P(v_k \mid q_t = S_i), \quad 1 \le i \le N, 1 \le k \le M$
5. The initial state distribution
   $\pi_i = P(q_1 = S_i), \quad 1 \le i \le N$

In compact notation we use $\lambda = (A, B, \pi)$; parameter values *N* (number of states) and *M* (number of symbols in observation alphabet) are not used explicitly here.

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

The Basic Problems for HMM

1. Given the observation sequence $O = O_1 O_2 \ldots O_T$ and model $\lambda = (A, B, \pi)$ compute the probability of the sequence, given the model $P(O|\lambda)$.
   Forward-Backward procedure
2. Given the observation sequence O and model $\lambda = (A, B, \pi)$ compute state sequence $Q = q_1 q_2 \cdots q_T$ which "best" explains the observations.
   Viterbi algorithm
3. Adjust model parameters $\lambda = (A, B, \pi)$ to maximize $P(\{O\}|\lambda)$ for observations set $\{O\}$.
   Baum-Welsh procedure

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Computing P(O|λ)

A trivial exhausting search has computational complexity of $O(2T*N^T)$.

Forward-Backward procedure iteratively computes total probability of the observation sequence *forward*:

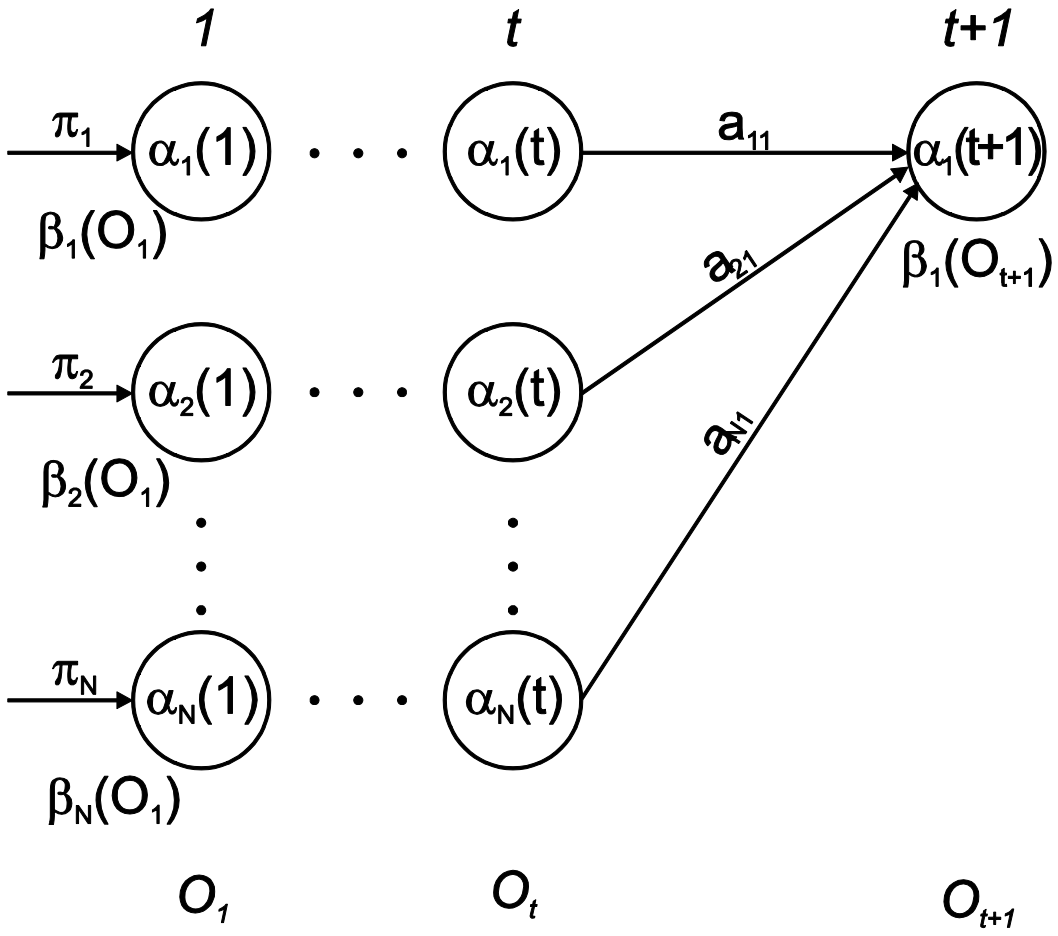
$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i | \lambda), 1 \le i \le N, 1 \le t \le T$$

and/or backward:

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

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Illustration of single step of the procedure



Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Forward procedure

1. Initialization

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

2. Induction

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

3. Termination

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

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Backward procedure

1. Initialization

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

2. Induction

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

3. Termination

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

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Viterbi Algorithm

Finding optimal state sequence $Q = q_1 q_2 \dots q_T$ for given observation sequence $O = O_1 O_2 \dots O_T$.

We define the variable $\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_t = S_i, O_1 O_2 \dots O_t | \lambda)$, which is the problem solution at time *t* and ends in state S_i. Trace variable (array) ψ will be used to backtrack state sequence, since δ describes only probability.

1. Initialization

$$\delta_1(i) = \pi_i b_i(O_1), 1 \le i \le N$$

$$\psi_1(i) = 0, 1 \le i \le N$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij} \right] b_j(O_t), 1 \le i \le N, 2 \le t \le T$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij} \right], 1 \le i \le N, 2 \le t \le T$$

3. Termination

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i) \right]$$

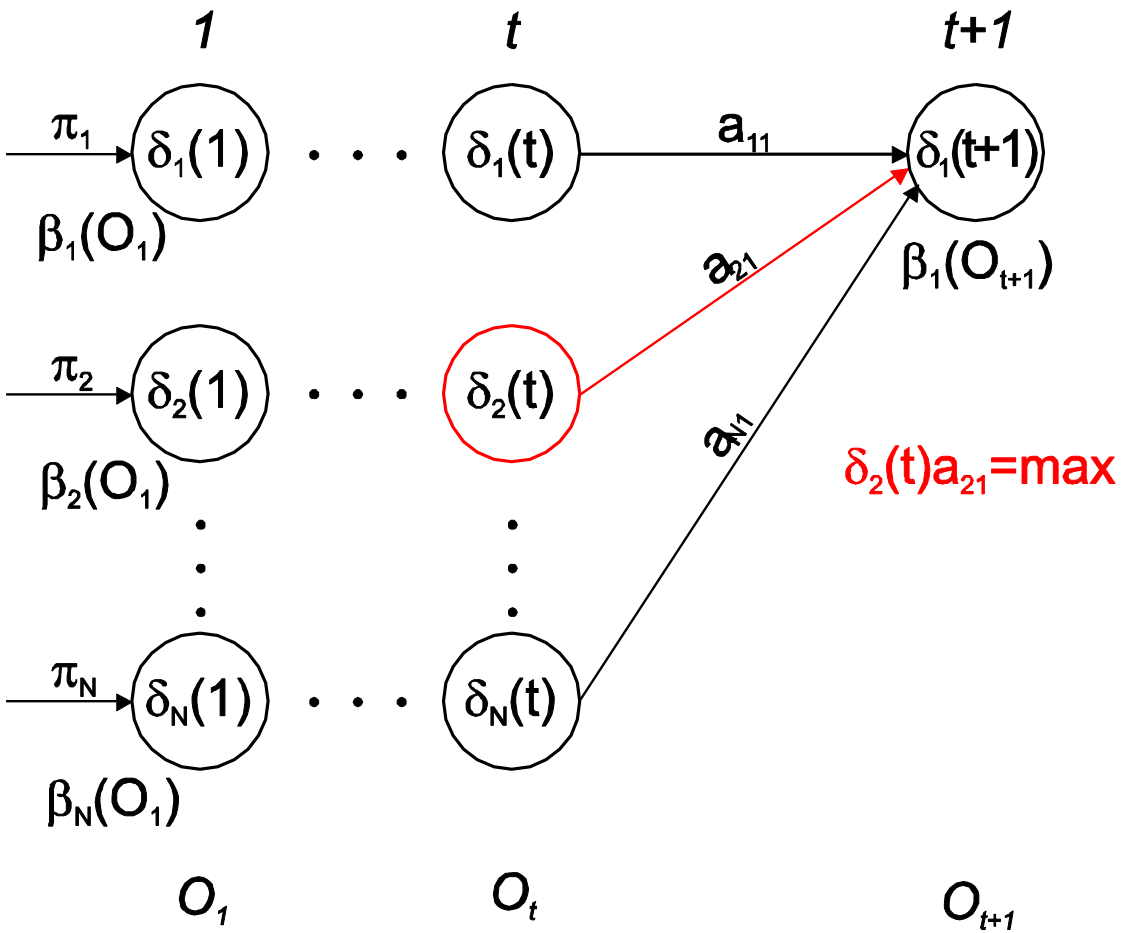
$$q_T^* = \arg\max_{1 \le i \le N} \left[\delta_T(i) \right]$$

4. State sequence backtracking

$$q_t^* = \psi_{t+1}\left(q_{t+1}^*\right), \quad t = T-1, T-2 \ldots 1$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Illustration of single step of Viterbi algorithm



(Generally it is the same as in *forward* procedure, but here we are only interested in one “best” transition to the given state)

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Baum-Welsh Algorithm

Computing model's parameters is not trivial problem. Difficulties start at the very beginning. It is relatively easy to assess the number of different observations, but the number of model's states is in general only the designer's presumption (excluding these rare cases when knowledge of the problem domain allows us to create reliable generative model).
Baum-Welsh algorithm relies on adjusting model's parameters to the observation sequence. We define auxiliary variable γ representing probability of transition from state *i* to *j* at time *t*:

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1)a_{ij}b_{jk}\beta_j(t)}{P(O \mid \lambda)}, \quad 1 \le i\ j \le N, 1 \le k \le M, 2 \le t \le T-1$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds

Baum-Welsh Algorithm

The value of variable γ is used to compute (better) estimates of model's parameters λ.
Estimate of the transition probability:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_{ij}(t)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \gamma_{ik}(t)}$$

Estimate of the probability of generating observable value:

$$\hat{b}_{ik} = \frac{\sum_{t=1}^{T} \sum_{l} \gamma_{ij}(t) \left| v(t) = v_k \right.}{\sum_{t=1}^{T} \sum_{l} \gamma_{il}(t)}$$

Preparing of these materials was supported by a grant from Liechtenstein, Iceland and Norway through EEA Grants and Norway Grants and co-financed by domestic funds