CS761 Artificial Intelligence

18. Probabilistic Reasoning over Time

Jiamou Liu
The University of Auckland

Probability and Time

Time is an important factor in many real-world problems:

- Predictions:

  e.g. (Trading agents) Predicting the market.

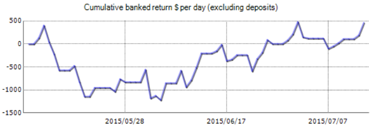
- Root cause analysis:

  e.g. Investigate the cause of a drink water contamination.

- Natural language generation:

  e.g. Choose the next word given the previously generated words in a sentence.

All these applications require a model that is able to represent time and how variables change with time.



Markov Chains

Definition

A Markov chain (MC) is a type of Bayesian network that is used to represent sequence of values; it consists of variables

$$S_0, S_1, S_2, \ldots, S_n$$

for some $n \in \mathbb{N}$ and directed edges $E = \{(S_t, S_{t+1}) \mid 0 \leq t < n\}$.

The MC is called stationary if $\mathbf{P}(S_{t+1} \mid S_t) = \mathbf{P}(S_{\ell+1} \mid S_\ell)$ for any $t, \ell \geq 0$.

Note:

- A variable S_t is not necessarily Boolean, i.e., S_t may have an arbitrary domain $\mathrm{dom}(S_t)$.
- All variables have the same domain, i.e., $\mathrm{dom}(S_0) = \mathrm{dom}(S_1) = \ldots = \mathrm{dom}(S_n)$.

Thus a stationary MC represents a special type of stochastic process:

- A stochastic process describes the evolution of some system.
- Each element in the domain $\mathrm{dom}(S_t)$ is called a state of the system.
- At each time step t, the system is at a particular state.
- If the process moves from state q to state q', then we say that the process made a transition.
- $\mathbf{P}(S_0)$ specifies the initial conditions.
- $\mathbf{P}(S_{t+1} \mid S_t)$ specifies transition probabilities.

Let C be a stationary Markov chain and $\mathrm{dom}(S_t) = \{q_1, q_2, \ldots, q_n\}$.

- Each probability distribution $\mathbf{P}(S_t \mid e)$ can be described by a vector (where e is any evidence)

$$\big(P(S_t = q_1 \mid e), P(S_t = q_2 \mid e), \ldots, P(S_t = q_n \mid e)\big)$$

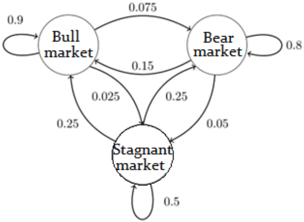
- The transition probability distribution $\mathbf{P}(S_{t+1} \mid S_t)$ can be represented by a transition matrix M_C, a $n \times n$ matrix where

$$M_C(i, j) = \mathbf{P}(S_{t+1} = q_j \mid S_t = q_i).$$

Example. [market] We can view the evolution of a market as a stochastic process, as shown in the diagram below.

A stationary MC model: Each variable S_t represents the state of the market at time t, where $\mathrm{dom}(S_t) = \{bull, bear, stagnant\}$, and

$$P(S_{t+1} = bull \mid S_t = bull) = 0.9,$$
$$P(S_{t+1} = bear \mid S_t = bull) = 0.075,$$
$$P(S_{t+1} = bear \mid S_t = stagnant) = 0.025,$$
$$P(S_{t+1} = bull \mid S_t = bear) = 0.15, \ldots$$



$$Mc = \begin{pmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

Possible query: What is the probability of bull market in 3 time steps if we start at a bull market?

Predicting Future States

- We can use any Bayesian network inference method (e.g. VE) on an MC.
- However, due to the sequential structure of MCs, inference can be solved more conveniently.
- Consider an MC C with 3 states.

$$\mathbf{P}(S_0) = (a_1, a_2, a_3) \qquad M_C = \mathbf{P}(S_{t+1} \mid S_t) = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

Suppose we want to eliminate S_0. Recall the VE algorithm:

1. Multiply the tables $\mathbf{P}(S_1 \mid S_0)$ and $\mathbf{P}(S_0)$ to get a new table $f(S_0, S_1)$
2. Sum-out S_0 from $f(S_0, S_1)$.

Example [rethinking VE]. Eliminating the variable S_0:

1. **Multiplication:** Treat $\mathbf{P}(S_0)$ and $\mathbf{P}(S_1 \mid S_0)$ as CPTs.

$$\mathbf{P}(S_0) = (a_1, a_2, a_3) \qquad \mathbf{P}(S_1 \mid S_0) = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

The table obtained from multiplication:

$$f(S_0, S_1) = \begin{pmatrix} a_1 b_{1,1} & a_1 b_{1,2} & a_1 b_{1,3} \\ a_2 b_{2,1} & a_2 b_{2,2} & a_2 b_{2,3} \\ a_3 b_{3,1} & a_3 b_{3,2} & a_3 b_{3,3} \end{pmatrix}$$

2. S_0**-Sum** from $f(S_0, S_1)$ to get

$$\mathbf{P}(S_1) = \left(\sum_{i=1}^{3} a_i b_{i,1}, \sum_{i=1}^{3} a_i b_{i,2}, \sum_{i=1}^{3} a_i b_{i,3} \right) = \mathbf{P}(S_0) \times \mathbf{P}(S_1 \mid S_0) = \mathbf{P}(S_0) M_C$$

Note. There is no need to normalise the resulting vector.

Example [rethinking VE]. Eliminating the variable S_0:

1. **Multiplication:** Treat $\mathbf{P}(S_0)$ and $\mathbf{P}(S_1 \mid S_0)$ as CPTs.

$$\mathbf{P}(S_0) = (a_1, a_2, a_3) \qquad \mathbf{P}(S_1 \mid S_0) = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

The table obtained from multiplication:

$$f(S_0, S_1) = \begin{pmatrix} a_1 b_{1,1} & a_1 b_{1,2} & a_1 b_{1,3} \\ a_2 b_{2,1} & a_2 b_{2,2} & a_2 b_{2,3} \\ a_3 b_{3,1} & a_3 b_{3,2} & a_3 b_{3,3} \end{pmatrix}$$

2. S_0**-Sum** from $f(S_0, S_1)$ to get

$$\mathbf{P}(S_1) = \left(\sum_{i=1}^{3} a_i b_{i,1}, \sum_{i=1}^{3} a_i b_{i,2}, \sum_{i=1}^{3} a_i b_{i,3} \right) = \mathbf{P}(S_0) \times \mathbf{P}(S_1 \mid S_0) = \mathbf{P}(S_0) M_C$$

Note. There is no need to normalise the resulting vector.

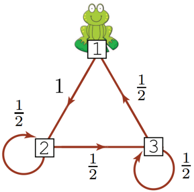
Theorem.

Let C be an n-state stationary MC. Then for any $1 \leq t \leq n$, the probability distribution

$$\mathbf{P}(S_t) = \mathbf{P}(S_0) M_C^t.$$

Example. [leaping frog] A frog hops about on 3 lily pads. The numbers next to arrows show the probabilities with which, at the next jump, he jumps to a neighbouring lily pad.

A stationary MC model:



$$M_C = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

Predictions.

$$\mathbf{P}(S_2) = \mathbf{P}(S_0)\,(M_C)^2 = (1,0,0)\begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}^2 = \left(0, \frac{1}{2}, \frac{1}{2}\right)$$

$\mathbf{P}(S_5) = (3/16, 13/32, 23/32)$, $\quad\mathbf{P}(S_7) \approx (0.203, 0.390, 0.406)$
$\mathbf{P}(S_{10}) = (0.19921875, 0.400390625, 0.400390625)$
$\mathbf{P}(S_{20}) \approx (0.200000007, 0.3999996, 0.3999996)$

Note: $(0.2, 0.4, 0.2)M_C = (0.2, 0.4, 0.2)$. So in the long run the frog's location distribution is $(0.2, 0.4, 0.4)$.

Stationary Distribution

Stationary Distribution

A stationary distribution $\mathbf{s}$ of a Markov chain is a stochastic vector such that

$$\mathbf{s}M_C = \mathbf{s}$$

Note:

- The stationary distribution does not depend on the initial vector
- A stationary distribution is an eigenvector of P with eigenvalue 1

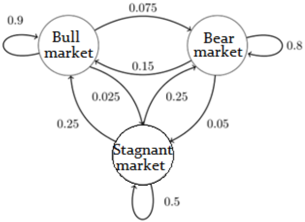
Theorem [Converging behaviour of MC]

Suppose a Markov chain is

1. strongly connected (i.e., the network (V, E) contains only one strongly connected component)
2. $P(S_{t+1} = v \mid S_t = v) > 0$ for any value $v \in \mathrm{dom}(S_t)$

Then the stochastic process of this Markov chain converges to a unique stationary distribution.

Example. [market] Suppose the market starts from Bull market. What is the long-term behaviour of the market?



$$Mc = \begin{pmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

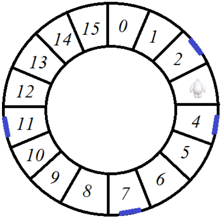
$$\mathbf{P}(S_0) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \to \begin{pmatrix} 0.9 \\ 0.075 \\ 0.025 \end{pmatrix} \to \begin{pmatrix} 0.8275 \\ 0.13375 \\ 0.03875 \end{pmatrix} \to \begin{pmatrix} 0.7745 \\ 0.17875 \\ 0.04675 \end{pmatrix} \to$$

$$\begin{pmatrix} 0.736 \\ 0.212775 \\ 0.051225 \end{pmatrix} \to \begin{pmatrix} 0.7071225 \\ 0.23822625 \\ 0.05465125 \end{pmatrix} \rightsquigarrow_{\infty} \begin{pmatrix} 0.625 \\ 0.3125 \\ 0.0625 \end{pmatrix}$$

Hidden Markov Model

- An MC models a dynamical system which evolves with time.
- Most of the time, we are observers of this system with incomplete/inprecise information about its internal states.
- Given the sequence of observations, we would like to reason about the internal states of the system.
- Therefore we need a model that extends MC with observations and hidden variables.

Example. [localisation] Imagine a robot exploring a circular corridor; it could be in one of 16 locations.



- Some positions $\{2, 4, 7, 11\}$ have a door.
- The robot can (noisily) sensor whether it is in front of a door.
  $P(Obs = door \mid door) = 0.8$, $P(Obs = door \mid \neg door) = 0.1$
- The robot can move left, right (with uncertainty) or stay still.
  $P(Loc_{t+1} = i \mid Act_t = right \text{ (or } left\text{)}, Loc_t = i) = 0.1$

  $P(Loc_{t+1} = i + 1 \mid Act_t = right \text{ (or } left\text{)}, Loc_t = i) = 0.8$

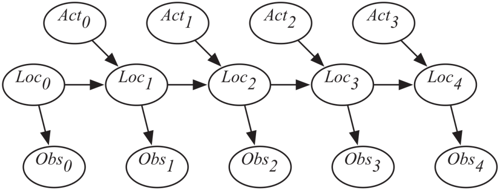
  $P(Loc_{t+1} = i + 2 \mid Act_t = right \text{ (or } left\text{)}, Loc_t = i) = 0.074$

  $P(Loc_{t+1} = j \mid Act_t = right \text{ (or } left\text{)}, Loc_t = i) = 0.002$ for any other j

The robot starts at an unknown location and must identify its location.

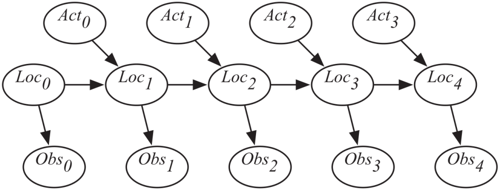
Definition

A hidden Markov model (HMM) is a Bayesian network that extending an MC by observation variables $O_0, O_1, \ldots$ (having the same domain), and (possibly) action variables $A_0, A_1, \ldots$ (having the same domain); the directed graph representation is of the form:



Definition

A hidden Markov model (HMM) is a Bayesian network that extending an MC by observation variables $O_0, O_1, \ldots$ (having the same domain), and (possibly) action variables $A_0, A_1, \ldots$ (having the same domain); the directed graph representation is of the form:



Example. [localisation]

- Hidden variable: L_t: the locations of the robot.
- Observed variabls: O_t: door or no door at time t, and A_t: action at time t (left,right,stay).

The problem (filtering or belief-state monitoring) is to compute the current internal state given a sequence of actions and observations.

The Filtering Problem

Input: Probabilities $\mathbf{P}(L_0)$, $\mathbf{P}(L_{t+1} \mid L_t, A_{t-1})$, $\mathbf{P}(O_t \mid L_t)$.
Observations: $o_0, \ldots, o_s$
Actions: $a_0, \ldots, a_{s-1}$
Goal: Compute $\mathbf{P}(L_s \mid o_0, a_0, o_1, a_1, \ldots, a_{s-1}, o_s)$.

The Filtering Problem

Input: Probabilities $\mathbf{P}(L_0)$, $\mathbf{P}(L_{t+1} \mid L_t, A_{t-1})$, $\mathbf{P}(O_t \mid L_t)$.
Observations: $o_0, \ldots, o_s$
Actions: $a_0, \ldots, a_{s-1}$
Goal: Compute $\mathbf{P}(L_s \mid o_0, a_0, o_1, a_1, \ldots, a_{s-1}, o_s)$.

Solution: We inductively compute the wanted probability.

- Base case: Suppose $s = 0$. Then we want to find $P(L_0 \mid o_0)$.

$$\mathbf{P}(L_0 \mid o_0) =\propto \mathbf{P}(o_0 \mid L_0)\mathbf{P}(L_0)$$

- Inductive step: Suppose $s > 0$.

We use $\overline{\mathbf{a}}_i$ to denote $a_0 \wedge a_1 \wedge \cdots \wedge a_i$ for any $i \in \mathbb{N}$.
We use $\overline{\mathbf{o}}_i$ to denote $o_0 \wedge o_1 \wedge \cdots \wedge o_i$ for any $i \in \mathbb{N}$.

Suppose we have already computed $\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})$.

Our goal is to compute $\mathbf{P}(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1})$.

$$
\begin{aligned}
&\mathbf{P}(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1}) \\
&\propto \mathbf{P}(L_s \wedge \overline{\mathbf{o}}_s \mid \overline{\mathbf{a}}_{s-1}) \\
&= \mathbf{P}(o_s \wedge (L_s \wedge \overline{\mathbf{o}}_{s-1}) \mid \overline{\mathbf{a}}_{s-1}) \\
&= \mathbf{P}(o_s \mid (L_s \wedge \overline{\mathbf{o}}_{s-1}) \wedge \overline{\mathbf{a}}_{s-1}) \times \mathbf{P}(L_s \wedge \overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-1}) \\
&= \mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \mathbf{P}(L_s \wedge L_{s-1} \wedge \overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-1}) \qquad \text{(by Law of Total Prob.)} \\
&= \mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-1})\mathbf{P}(L_{s-1} \wedge \overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-1})\Big] \\
&= \mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge \overline{\mathbf{a}}_{s-1})\mathbf{P}(L_{s-1} \wedge \overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-2})\Big] \\
&= \mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\mathbf{P}(\overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-2})\Big] \\
&= \mathbf{P}(\overline{\mathbf{o}}_{s-1} \mid \overline{\mathbf{a}}_{s-2})\mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\Big] \\
&\propto \mathbf{P}(o_s \mid L_s) \times \sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\Big]
\end{aligned}
$$

We use two operations to realise this formula:

$$\mathbf{P}(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1}) \propto \underbrace{\mathbf{P}(o_s \mid L_s)\times}_{Op2} \underbrace{\sum_{L_{s-1}} \left[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\right]}_{Op1}$$

We use two operations to realise this formula:

$$\mathbf{P}(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1}) \propto \underbrace{\mathbf{P}(o_s \mid L_s)\times}_{Op2} \underbrace{\sum_{L_{s-1}} \Big[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\Big]}_{Op1}$$

Op 1. Suppose $P(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2}) = (a_1, a_2, a_3)$,

and $P(L_s \mid L_{s-1} \wedge a_{s-1}) = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$.

So the result should be

$$(c_1, c_2, c_3) = (a_1, a_2, a_3) \times \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

We use two operations to realise this formula:

$$\mathbf{P}(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1}) \propto \underbrace{\mathbf{P}(o_s \mid L_s)\times}_{Op2} \underbrace{\sum_{L_{s-1}} \left[\mathbf{P}(L_s \mid L_{s-1} \wedge a_{s-1})\mathbf{P}(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2})\right]}_{Op1}$$

Op 1. Suppose $P(L_{s-1} \mid \overline{\mathbf{o}}_{s-1} \wedge \overline{\mathbf{a}}_{s-2}) = (a_1, a_2, a_3)$,

and $P(L_s \mid L_{s-1} \wedge a_{s-1}) = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$.

So the result should be

$$(c_1, c_2, c_3) = (a_1, a_2, a_3) \times \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}$$

Op 2. Suppose $P(o_s \mid L_s) = (d_1, d_2, d_3)$. Then the result should be

$$P(L_s \mid \overline{\mathbf{o}}_s \wedge \overline{\mathbf{a}}_{s-1}) \propto (d_1 c_1, d_2 c_2, d_3 c_3)$$

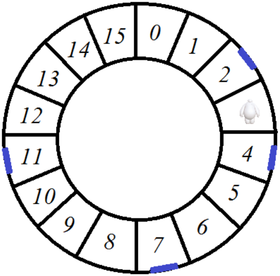
Forward algorithm for the Filtering Problem of HMM

INPUT: Initial condition $P(L_0)$, transition matrices $P(L_{t+1} \mid L_t, A_t)$, and $P(O_t \mid L_t)$. Observations $o_0, o_1, \ldots, o_s$. Actions $a_0, a_1, \ldots, a_s$. $s \in \mathbb{N}$

OUTPUT: $P(L_s \mid o_0 \wedge a_0 \wedge \cdots \wedge a_{s-1} \wedge o_s)$

1. $n := |\mathrm{dom}(L_0)|$
2. Initialise a length-n vector $\vec{r} = (r_1, r_2, \ldots, r_n) := P(L_0)$
3. for each i in $\{0..s\}$ do
4. $\vec{r} := \vec{r} \times P(L_{i+1} \mid L_i, a_i)$
5.   Say $P(o_i \mid L_t) = (d_1, d_2, \ldots, d_n)$
6. $\vec{r} = (r_1, \ldots, r_n) := (r_1 d_1, r_2 d_2, \ldots, r_n d_n)$
7.   Normalise $\vec{r}$ so that all entries sum to 1
8. Return $\vec{r}$

Exercise: Solve the localisation problem described above.



Summary of The Topic

The following are the main knowledge points covered:

- **Markov chain:** A Bayesian network with variables $S_0, S_1, \ldots, S_n$.
- **Stationary Markov chain:**
  - $\mathbf{P}(S_{t+1} \mid S_t) = \mathbf{P}(S_{\ell+1} \mid S_\ell)$ for any $t, \ell \geq 0$.
  - Transition matrix $M_C = \mathbf{P}(S_{t+1} \mid S_t)$.
- **Predicting future states:** $\pi(S_i) = \mathbf{P}(S_0)M_C^i$
- **Stationary distribution:** $\mathbf{s}M_c = \mathbf{s}$
- **Hidden Markov model:** Extending MC with observations and hidden variables.
- **Filtering problem:** Given HMM, compute $\mathbf{P}(L_s \mid o_0, a_0, o_1, a_1, \ldots, a_{s-1}, o(s))$.
- **Forward algorithm for the filtering problem of HMM**