A Markov chain is given by:

- A set of states $S = \{S_1, \ldots, S_N\}$
- A transition probability matrix $P = \{p_{ij}\}$

$p_{ij}$ represents the probability of transitioning from $S_i$ to $S_j$. 

$\begin{array}{cccc}
S_1 & S_2 & \cdots & S_N \\
S_1 & 1 & & \\
S_2 & & \ddots & \\
S_N & & & \end{array}$
ELEMENTS OF AN HMM

1) \( N = \text{number of states} \quad S = \{S_1, \ldots, S_N\} \)

2) \( M = \text{number of observation symbols emitted by a state} \)
   \( V = \{v_1, v_2, \ldots, v_M\} \) alphabet

3) \( A = \{a_{ij}\} \) the transition probability distribution

\[ a_{ij} = P[q_t = S_j \mid q_{t+1} = S_i, q_t = S_i, 1 \leq i \leq N, q_t = \text{state at time } t, 1 \leq i \leq N, \text{ time steps } t = 1, 2, \ldots] \]
4) \( B = \frac{1}{J} \delta_{j,k} \) the observation symbols probability distribution

\( b_{j,k} = P(\text{emitting symbol } y_k \text{ at time } t | \varphi = s_j) \)

state \( s_j \) emits \( y_k \) at time \( t \)

5) \( \pi = \frac{1}{N} \pi_i \) initial state distribution \( \leq i \leq N \)

\( \pi_i = P(\text{starting in state } s_i) \)

HMM Model \( \lambda = (A, B, \pi) \)

\( A N \times N \)

\( B N \times M \)

\( \pi N \times 1 \)

LABINER TUTORIAL Speech Recognition
3 FUNDAMENTAL PROBLEMS FOR HMMs

PB1. THE EVALUATION PB

Given: \( \lambda = (A, B, \pi) \), and an observation sequence \( \Theta = \sigma_1 \sigma_2 \ldots \sigma_T \), \( \sigma_i \in \Sigma \) alphabet

Compute: \( P(\Theta | \lambda) \) = the probability of observing the observation sequence \( \Theta \) in the model \( \lambda \).

PB2. THE DECODING PB

Decoding the "HIDDEN" part
Given: $\lambda = (A, B, \pi), \Theta = \theta_1, \ldots, \theta_T$

\underline{Compute: A sequence of states $Q = q_1, \ldots, q_T$ that gives the "best explanation" of the observations (most likely path of) $T$ states.}

Highest probability path

The Viterbi algorithm gives the solution to PB.2

PB.3 The learning PB

\underline{Given:} $\Theta = \theta_1, \ldots, \theta_T$

\underline{Compute:} $\lambda = (A, B, \pi)$

Best model $\approx \max \{ P(\Theta | \lambda) \}$
Given \[ \text{SOL TO PB 1.} \]
\[ \text{SOL TO PB 2.} \]

Dynamic Programming Alg \[ O(N^2 T) \]

Exact solutions in polynomial time

SOL TO PB 3 is not available
SOL: approximate maximum likelihood

(CSCI 1820: Spring 2022)

EM

HMM as a sequence generator
Given \( N, M, A, B, \pi \) the HMM can be used as a generator to give an observation sequence as output.

\[ \Theta = o_1 o_2 \ldots o_T \]

Length \( T \)

where each \( o_i \in \Sigma \).

\( T = \# \) of observations in \( \Theta \).

Observation \( = \) symbol/letter
Algorithm:

1. Choose an initial state $x_1 = S_i$

   according to initial state distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$

   $\pi_i$ is the probability of initial state to be $S_i$, $1 \leq i \leq N$

2. Set $t = 1$

Random number $r \in [0, 1]$
③ Choose $\xi_t = \nu_k$ according to symbol probability distribution for state $S_i$ emitted symbol

\[ S_i \xrightarrow{a, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}} S_j \]

④ Transit to new state

$\xi_{t+1} = S_j$ according to probability state transition matrix

$\alpha_{ij}^{t+1} : S_i$
(5) Set \( t = t + 1 \), return to Step 3 if \( t < T \), otherwise terminate.

\[ S_{10}, S_{25}, S_1, S_3, S_{100} \]
\[ \Theta = \begin{array}{cccc}
\alpha & \alpha & \beta & \beta \\
\end{array} \]

The above algorithm can be used as a generator of observation sequences, but also as a model \( \lambda = (A, \Pi) \) for a class of observation sequences for this HMM.
Solution to PB1

Given \( \lambda \)

Compute \( P(O|\lambda) = \)

= the probability of observing \( O \) in \( \lambda \)

PB.1 "Evaluation Pb" or "Model scoring Pb"

\( \lambda_1 > \lambda_2 \) two models for the same application which model is better?

\( P(O|\lambda_1) > P(O|\lambda_2) \)

\( \lambda_1 \) is a better model
Sol to PB1

\[ \theta = \sigma_1 \sigma_2 \ldots \sigma_T \uparrow \uparrow \uparrow \]

\[ Q = q_1, q_2, \ldots, q_T \]

\[ \lambda = (A, B, \alpha) \]

\[ \gamma_i = \text{initial state} \]

(1) Thus compute:

\[ P(\theta | Q) = \prod_{i=1}^{T} P(\sigma_i | q_i) = \]

(2) Let us compute:

\[ b_{q_i}(\sigma_i) \]
\[ Q = a_1 a_2 \ldots a_T \]

\[ P(Q) = \prod_{q_1, q_2, \ldots, q_T} a_{q_1} a_{q_2} a_{q_T} \]

Let us compute:

\[ P(O, Q) \text{ Joint probability of } O \text{ and } Q \text{ occurring together} \]

\[ P(O, Q) = P(O \mid Q) \cdot P(Q) \]

Standard prob theory:

"Conditional prob"

\[ P(O \mid Q) = \sum_{all \ Q} P(O \mid Q) \cdot P(Q) = \]

Objective function
\[ \sum \frac{b(\Theta_1) b(\Theta_2) \cdots b(\Theta_T)}{2_1 2_2 \cdots 2_T} \pi_{a_1} a_{q_1} a_{q_2} \cdots a_{q_T} \]

\[ \sum \prod_{t=1}^{T} \frac{b(\Theta_t) a_{q_t} b(\Theta_{t+1}) a_{q_{t+1}} b(\Theta_{t+2}) a_{q_{t+2}} \cdots b(\Theta_T)}{2_1 2_2 \cdots 2_T} \]

\[ \cdots a_{q_T} a_{q_{T-1}} \frac{b(\Theta_T)}{2_T} \]

**But this is an \( O(N^T) \) exponential deg.**

\[ \text{ex.} \quad N = 50 \quad T = 1000 \quad 1000 \text{ is more than all atoms} \]
Can we do better? YES \( O(N^{2T}) \)

\( 50 \times 50 \times 1000 \)

poly time, practical

---

The FORWARD VARIABLE

forward variable \( \alpha_t(i) \)

\[
\alpha_t(i) = \frac{P(o_1, o_2, \ldots, o_t) \cdot \pi_i}{Z_t}
\]

i.e. the probability of the partial observation \( o_1, o_2, \ldots, o_t \)

until time \( t \) and being
in state $s_i$ at time $t$

**THE FORWARD ALGORITHM**

**Input:** $\Theta \rightarrow \lambda = (A, B, \pi)$

**Output:** $P(\Theta | \lambda)$

1. **Initialization**

   $$\alpha_1(i) = \pi_i \cdot b(\sigma_1)$$
   $$1 \leq i \leq N$$

2. **Recurrence**

   $$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \cdot b(\sigma_{t+1})$$
   $$1 \leq i \leq N$$
   $$1 \leq t \leq T-1$$
PB.2 \[ \text{MAX} \quad 1 \leq j \leq N \]

3) \text{TERMINATION}

\[ P(\theta | \lambda) = \sum_{i=1}^{N} \lambda^i (1 - \lambda) \]

\text{time} \quad O(n^{2/3})

\[ q_1, q_2, \ldots, q_T \]

Diagram with elements labeled...