Hidden Markov Model

Tran Thi Hong Minh

Content

- Markov chain
- Hidden Markov model (HMM)
- Three problems of HMM

Discrete Markov chain

- Set of **state S** = $\{s_1, s_2, \ldots, s_N\}$. N is the number of states (s1=Sunny; s2=Rainy).

Eg: S = {Sunny, Rainy}

- Regularly spaced discrete times: t = 1,2,…

- The **initial state distribution π (= prior probability)** where π_i represents the probability that the process begin in state s_i. Eg: π = (0.2, 0.8)

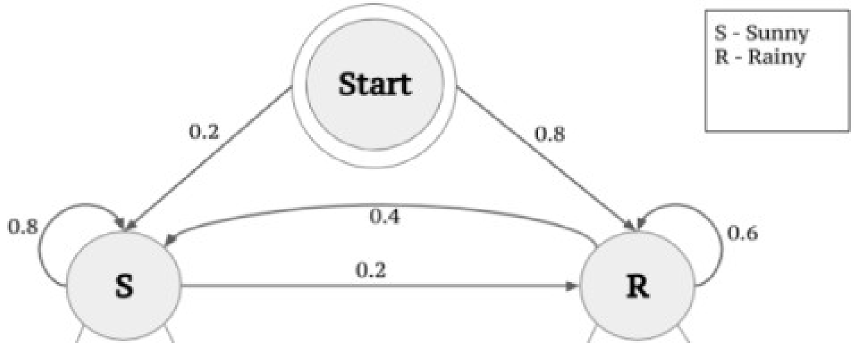
- Set **Q** = $\{q_1, q_2, \ldots, q_T\}$

q_t is a state at time point t

Eg:

s1→ s1 → s2 → s1 → s2 → s2

q1→ q2 → q3 → q4 → q5 → q6



Discrete Markov chain

- Future state q_t only depends on present state q_{t-1}, not relevant to any further past state (q_{t-2}, q_{t-3}, …, q_1).

$$P[q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k \ldots] = P[q_t = S_j \mid q_{t-1} = S_i]$$

- **Transition probability matrix A** and **transition probability distribution a_{ij}** $\quad a_{ij} \geq 0, \quad \sum_{j=1}^{N} a_{ij} = 1$

$$1 \leq i, j \leq N$$

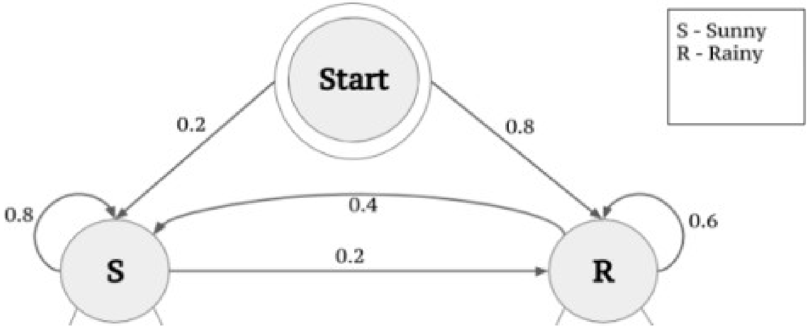
$A = a_{ij} =$

	S	R
S	0.8	0.2
R	0.4	0.6

- Given sunny at t=1:

What is the probability that the weather for the next 5 days will be sunny-rainy-sunny-rainy-rainy?

What is P(Q|Model) ?

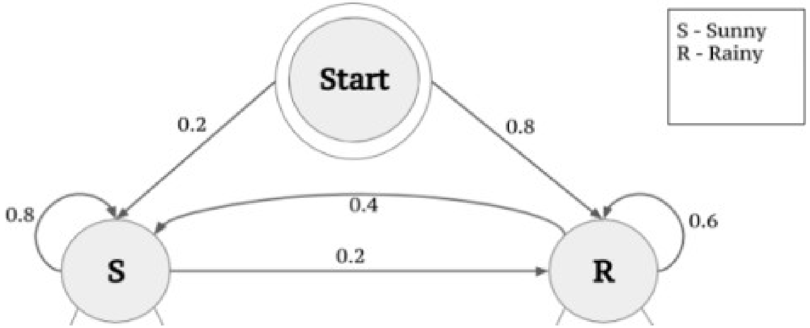


$P(O=\{s_1,s_1,s_2,s_1,s_2,s_2\} \mid A,\ q_1=S_1)$

$P(O=\{S,S,R,S,R,R\} \mid A,\ q_1=S_1)$

$= \pi \times a_{11} \times a_{12} \times a_{21} \times a_{12} \times a_{22}$

$= 0.2 \times 0.8 \times 0.2 \times 0.4 \times 0.2 \times 0.6$



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

Hidden Markov Model (HMM)

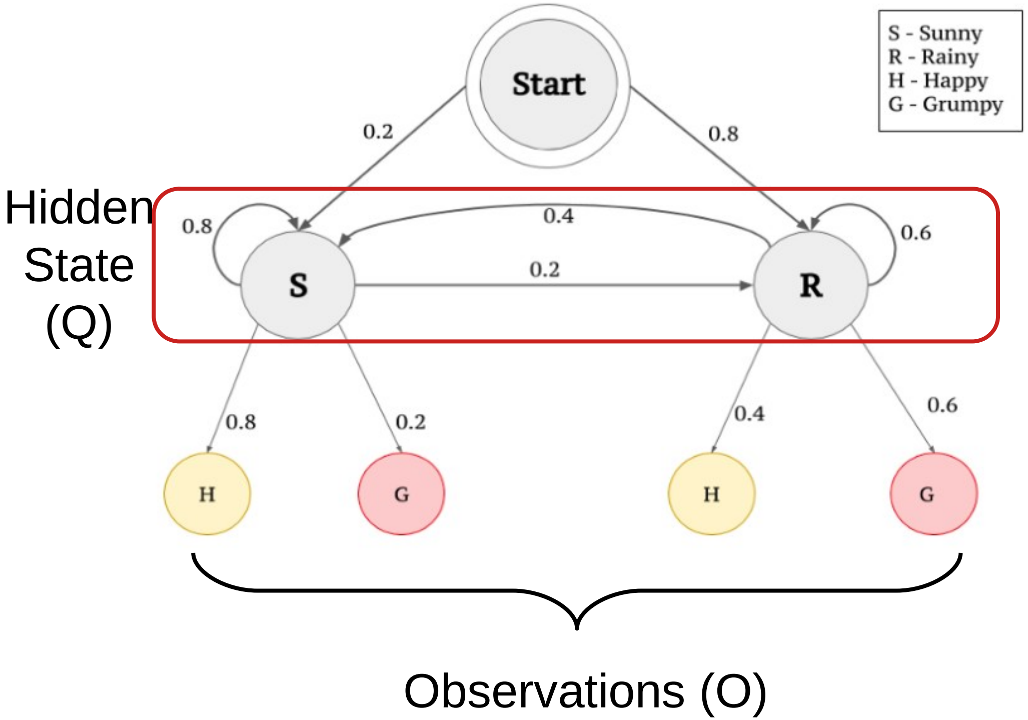
- There are a lot of cases where we can't observe the state (S) that we are interested in.
- We can only see the **output (observation O)**

$O = \{o_1, o_2, \ldots, o_T\}$

Eg:

$O = \{o_1, o_2, \ldots, o_T\}$

$= H, H, G, H, \ldots, G$



Hidden Markov Model (HMM)

A hidden Markov model has:

- N (hidden) states.

$$S = \{S_1, S_2, S_3, S_4, S_5, \ldots, S_N\}$$

state at time t is q_t $\quad \forall_{1 \le i \le t} i : q_i \in S$

- M, the number of observations (Happy, Grumpy)

$$V = \{V_1, V_2, V_3, V_4, V_5, \ldots, V_M\}$$

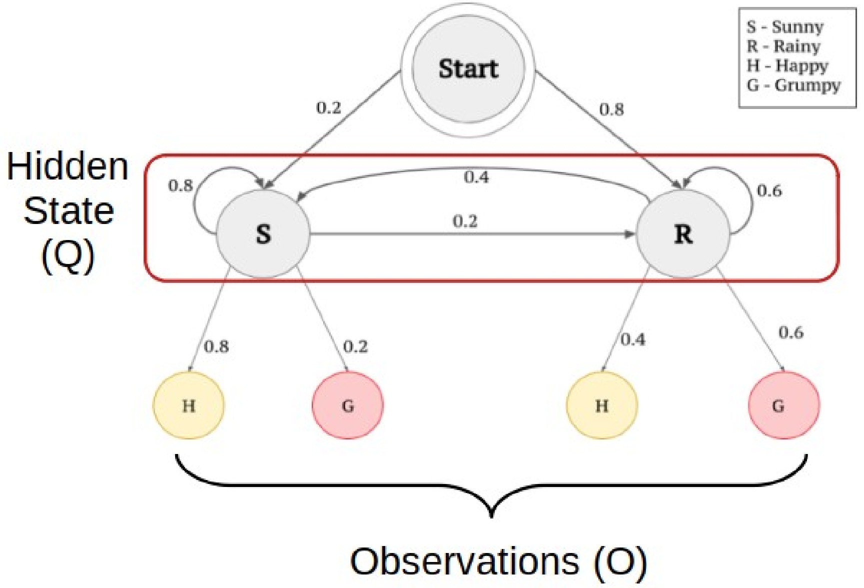
- State transition matrix A

$A = \{a_{ij}\}$

$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \quad 1 \le i, j \le N$

- if $a_{ij} = 0$ then a transition between S_i and S_j is not possible

$$\sum_{j=1}^{N} a_{ij} = 1$$



A =

	S	R
S	0.8	0.2
R	0.4	0.6

Hidden Markov Model (HMM)

A hidden Markov model has:

- **Emission probabilities (B) = Observation probabilities**

$$B = \{b_j(k)\}$$

$$b_j(k) = P(v_k \ at \ t | q_t = S_j)$$

$$1 \leq j \leq N$$
$$1 \leq k \leq M$$

$b_s(H) = 0.8$, $b_s(G) = 0.2$

$b_R(H) = 0.4$, $b_R(G) = 0.6$

- Initial probability distribution π

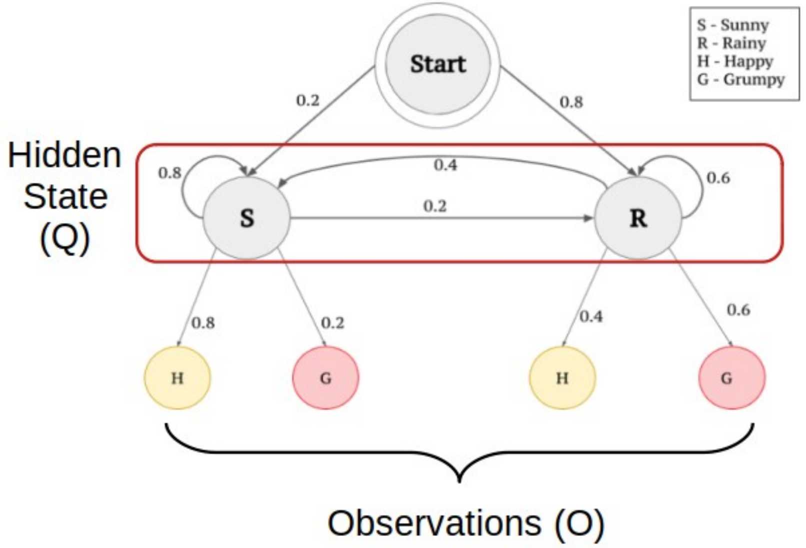
$$\pi = \{\pi_i\}$$

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N$$

- **Model λ** = (A,B,π)

$$O = \{O_1, O_2, O_3, \ldots, O_T\}$$

$$O_i \in V$$



Three problems of HMM

1. Given $\lambda=(A,B,\pi)$ and a sequence of observations $O = O_1O_2\ldots O_T$. Compute the probability that λ generated a sequence of observations, $P(O|\lambda)=?$

   Forward procedure, backward procedure

2. Given observation sequence $O = O_1O_2\ldots O_T$ and λ. What sequence of states ($Q = q_1q_2\ldots q_t$) best explains a sequence of observations

   Forward-backward algorithm, Viterbi

3. How to estimate $\lambda=(A,B,\pi)$ so as to maximize $P(O|\lambda)$ $\lambda = (A, B, \pi)$?

   Baum-Welch (Expectation maximization)

$$(A,B,\pi) = \underset{A,B,\pi}{\operatorname{argmax}}\ P(O|\lambda)$$

Problem 1

- Let's start by imagining all possible state sequences

$$Q = q_1, q_2, q_3, \ldots, q_T$$

$$O = \{O_1, O_2, O_3, \ldots, O_T\} \quad \lambda = (A, B, \pi)$$

  - Probability of seeing observations given those states is

$$P(O \mid Q, \lambda) = \prod_{t=1}^{T} P(O_t \mid q_t, \lambda)$$

$$P(O \mid Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T)$$

  - Probability of seeing those $\boxed{\text{state transitions}}$ is

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

- Probability of those seeing observations $\boxed{\text{and}}$ those state transitions is

$$P(O, Q \mid \lambda) = P(O \mid Q, \lambda) P(Q \mid \lambda)$$

- But we want the probability of the observations
- regardless of the particular state sequence

$$\boxed{\begin{aligned} P(O \mid \lambda) &= \sum_{all\ \ Q} P(O \mid Q, \lambda) P(Q \mid \lambda) \\ P(O \mid \lambda) &= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned}}$$

Problem 1

Given O = H H G H G H

- Suppose sequence Q = S R R S R R

$P(O|Q,\lambda) = b_S(H) \times b_R(H) \times b_R(G) \times b_S(H) \times b_R(G) \times b_R(H)$

$= 0.8 \times 0.4 \times 0.6 \times 0.8 \times 0.6 \times 0.4$

$P(Q|\lambda) = \pi_S a_{SR} a_{RR} a_{RS} a_{SR} a_{RR}$

$= 0.2 \times 0.2 \times 0.6 \times 0.4 \times 0.2 \times 0.6$

- Suppose sequence Q = R S R S S R

$P(O|Q,\lambda) = b_R(H) \times b_S(H) \times b_R(G) \times b_S(H) \times b_S(G) \times b_R(H)$

$= 0.4 \times 0.8 \times 0.6 \times 0.8 \times 0.2 \times 0.4$

$P(Q|\lambda) = \pi_R a_{RS} a_{SR} a_{RS} a_{SS} a_{SR}$

$= 0.8 \times 0.4 \times 0.2 \times 0.4 \times 0.8 \times 0.2$

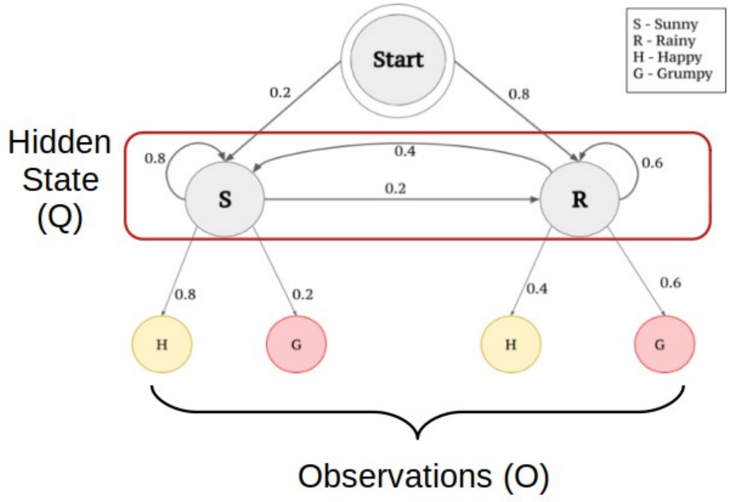
=> **Each given path Q has a probability for O**

=> **Each given path Q has its own probability**

$b_S(H) = 0.8, b_S(G) = 0.2$

$b_R(H) = 0.4, b_R(G) = 0.6$

	S	R
S	0.8	0.2
R	0.4	0.6



Problem 1

Therefore, total probability of O = H H G H G H

Sum over all possible paths Q: each Q with its own probability multiplied by the probability of O given Q

$$P(O \mid \lambda) = \sum_{all\ Q} P(O \mid Q, \lambda) P(Q \mid \lambda)$$

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

- Calculating this directly is infeasible

- How many state sequences are there? N^T
- How many multiplications per state sequence?

$$2T - 1$$

- Total number of operations?

$$(2T-1)N^T + (N^T - 1)$$

- T=100 and N=5, How many operations?

$$(2T-1)N^T + (N^T - 1)$$

$$(2(100)-1)5^{100} + (5^{100} - 1)$$

$$199 \cdot 5^{100} + 5^{100} - 1$$

$$200 \cdot 5^{100} - 1$$

$$\approx 5^{103}$$

$$\approx 10^{72}$$

N: the number of states;   T observations

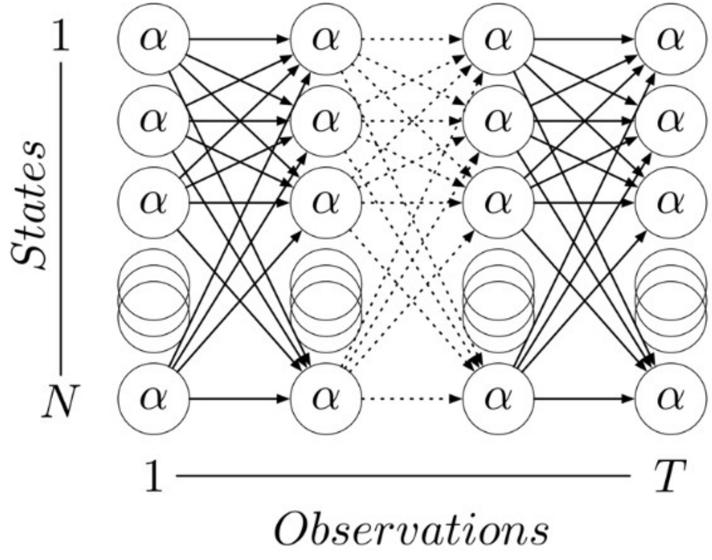
Solution for Problem 1: Forward procedure

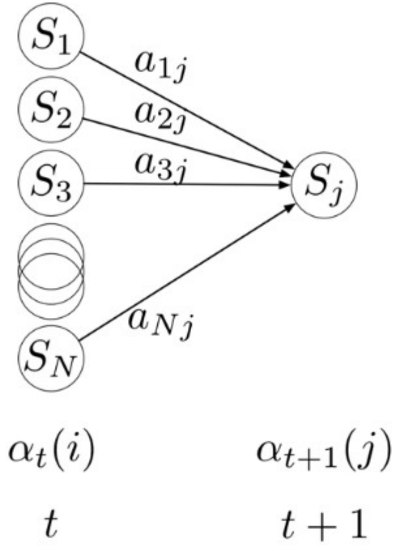
$$\alpha_t(i) = P(O_1, O_2, O_3, \ldots, O_t, q_t = S_i \mid \lambda)$$

The joint probability $\boldsymbol{\alpha_t(i)}$ is called **forward variable** at time point t and state s_i

$\alpha_t(i)$ is the probability of seeing observations O_1, O_2, …, O_t and then ending up in state s_i at time q_t given the model λ

α helps to reduce time and the number of repeated calculations, because it only considers all possible state sequences up to time t, not considering all possible path Q





Solution for Problem 1: Forward procedure

- base case: $\alpha_1(i) = \pi_i b_i(O_1) \quad 1 \le i \le N$
- inductive step:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad \begin{array}{l} 1 \le t \le T-1 \\ 1 \le j \le N \end{array}$$

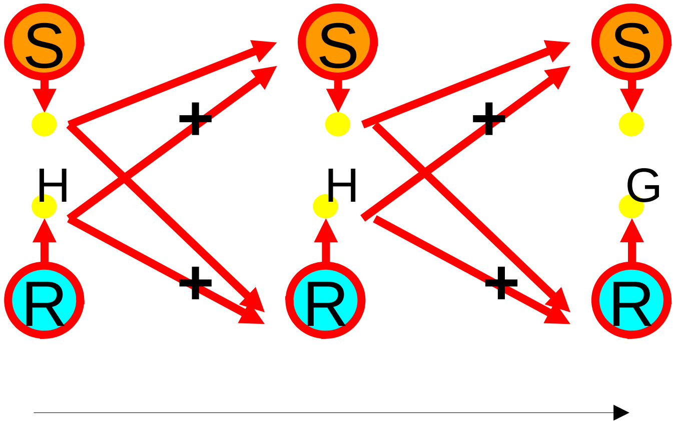
- final step:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)$$

O = H H G

$\alpha_1(S) = \pi_S b_S(H) = 0.2 \times 0.8 = 0.16$

$\alpha_1(R) = \pi_R b_R(H) = 0.8 \times 0.4 = 0.32$



$\alpha_2(S) = (\alpha_1(S) a_{SS} + \alpha_1(R) a_{RS}) b_S(H) = (0.16 \times 0.8 + 0.32 \times 0.4) \times 0.8 = 0.2048$

$\alpha_2(R) = (\alpha_1(S) a_{SR} + \alpha_1(R) a_{RR}) b_R(H) = (0.16 \times 0.2 + 0.32 \times 0.6) \times 0.4 = 0.0896$

Solution for Problem 1: Forward procedure

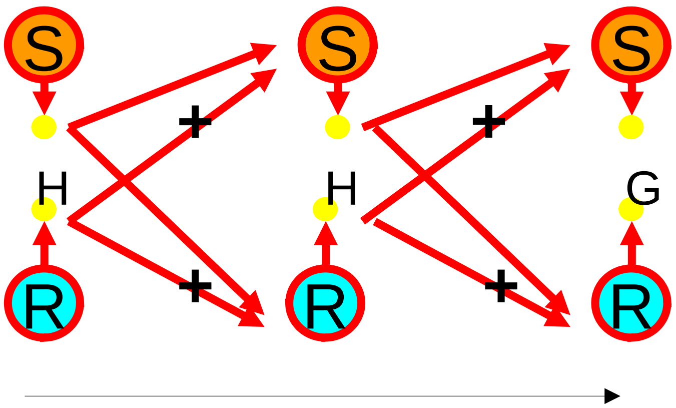
- base case: $\alpha_1(i) = \pi_i b_i(O_1) \qquad 1 \le i \le N$
- inductive step:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad \begin{array}{l} 1 \le t \le T-1 \\ 1 \le j \le N \end{array}$$

- final step:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)$$

O = H H G



Possible path Q:
S-S-S
S-S-R
S-R-R
S-R-S
R-R-R
R-S-S
R-S-R
R-R-S

$\alpha_3(S) = (0.2048 \times 0.8 + 0.0896 \times 0.4) \times 0.2 = 0.039936$

$\alpha_3(R) = (0.2048 \times 0.2 + 0.0896 \times 0.6) \times 0.6 = 0.056832$

$P(O|\lambda) = \alpha_3(S) + \alpha_3(R) = 0.096768$

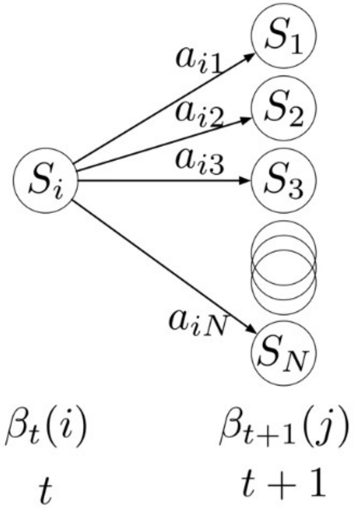
Solution for Problem 1: Backward procedure

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots O_T \mid q_t = Si, \lambda)$$

- base case: $\beta_T(i) = 1 \qquad 1 \le i \le N$
- inductive step:

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

$$t = T-1, T-2, \cdots, 1 \quad 1 \le i \le N$$



Final:

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

Both forward and backward could be used to solve problem 1, which should give identical results

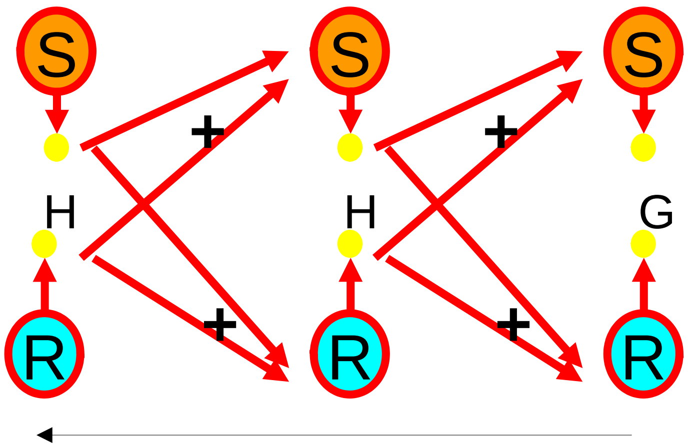
Solution for Problem 1: Backward procedure

- base case: $\beta_T(i) = 1 \qquad 1 \le i \le N$
- inductive step:

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

$$t = T-1, T-2, \cdots, 1 \quad 1 \le i \le N$$

O = H H G



Final:

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

$\beta_{T-1}(S) = a_{SS} b_S(G) \times 1 + a_{SR} b_R(G) \times 1 = 0.8 \times 0.2 + 0.2 \times 0.6 = 0.28$

$\beta_{T-1}(R) = a_{RS} b_S(G) \times 1 + a_{RR} b_R(G) \times 1 = 0.4 \times 0.2 + 0.6 \times 0.6 = 0.44$

$\beta_{T-2}(S) = a_{SS} b_S(H) \times 0.28 + a_{SR} b_R(H) \times 0.44 = 0.2144$

$\beta_{T-2}(R) = a_{RS} b_S(H) \times 0.28 + a_{RR} b_R(H) \times 0.44 = 0.1952$

$P(O|\lambda) = \Sigma_{i=1}^{N} \pi_i b_i(H) \beta_1(i) = 0.2 * 0.8 * 0.2144 + 0.8 * 0.4 * 0.1952 = 0.096768$

Three problems of HMM

1. Given λ=(A,B,π) and a sequence of observations O = $O_1O_2 \ldots O_T$. Compute the probability that λ generated a sequence of observations, $P(O|\lambda)$=?

    Forward procedure, backward procedure

2. Given observation sequence O = $O_1O_2 \ldots O_T$ and λ. What sequence of states (Q = $q_1q_2 \ldots q_t$) best explains a sequence of observations

    Forward-backward algorithm, Viterbi

3. How to estimate λ=(A,B,π) so as to maximize $P(O|\lambda)$ $\lambda = (A, B, \pi)$?

    Baum-Welch (Expectation maximization)

$$(A,B,\pi) = \underset{A,B,\pi}{\operatorname{argmax}}\ P(O|\lambda)$$

Problem 2

- "Go through all possible Q and pick the one leading to maximizing the criterion $P(Q|O,\lambda)$"

$$Q = \underset{q}{\operatorname{argmax}}(P(Q|O,\lambda))$$

Impossible if the number of states and observations is huge.

=> **forward-backward algorithm**

$\gamma_t(i)$ is the probability in state i at time q_t given observations and model λ = the probability in a particular position given all the observations that had come before and all the observations that are coming after + model λ

$\gamma_t(i)$ is also called individually optimal criterion

$$\xrightarrow{\alpha} \quad \xleftarrow{\beta}$$

O = H H G H G H

$$\gamma_t(i) = P(q_t = S_i \mid O, \lambda)$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

Problem 2: Forward-Backward algorithm

$\alpha_t(i)$ is the probability given regardless of the way that we got to state i at time t after seeing all observations up until time t

$\beta_t(i)$ is the probability starting in state i and we will see all remainder of observations up until time T

- Run forward α and backward β separately

- Keep track of the scores at every point

- Compute γ

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

- Determining optimal state q_t of Q at time point t to maximizes γt (i) over all values s_i

$$q_t = \underset{1 \leq i \leq N}{argmax}[\gamma_t(i)], \quad 1 \leq t \leq T$$

$\alpha_1(S) = \pi_S b_S(H) = 0.2 \times 0.8 = 0.16$

$\alpha_1(R) = \pi_R b_R(H) = 0.8 \times 0.4 = 0.32$

$\alpha 2\ (S) = 0.2048$

$\alpha_2(R) = 0.0896$

$\alpha_3(S) = 0.039936$

$\alpha_3(R) = 0.056832$

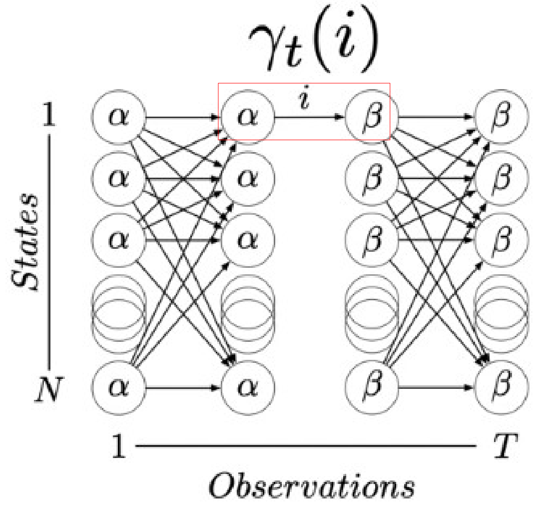
$\beta_3\ (S/R) = 1$

$\beta 2\ (S) = a_{SS} b_S(G) \times 1 + a_{SR} b_R(G) \times 1 = 0.28$

$\beta 2\ (R) = a_{RS} b_S(G) \times 1 + a_{RR} b_R(G) \times 1 = 0.44$

$\beta 1\ (S) = a_{SS} b_S(H) \times 0.28 + a_{SR} b_R(H) \times 0.44 = 0.2144$

$\beta 1\ (R) = a_{RS} b_S(H) \times 0.28 + a_{RR} b_R(H) \times 0.44 = 0.1952$



α

$\overrightarrow{}$

O = H H G

$\overleftarrow{}$

β

$\gamma 1\ (S) = 0.354$

$\gamma 1\ (R) = \boxed{0.646}$

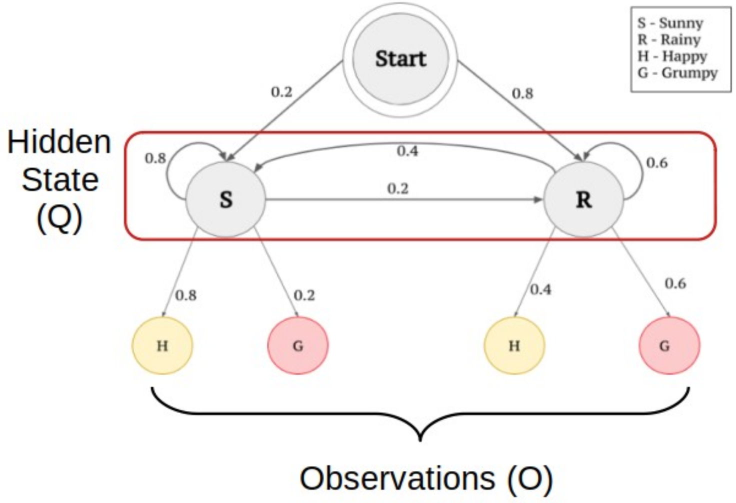
$\gamma 2\ (S) = \boxed{0.594}$

$\gamma 2\ (R) = 0.406$

$\gamma 3\ (S) = 0.412$

$\gamma 3\ (R) = \boxed{0.588}$

Consider γ in all states at every time t and choose the best one



As a result, the optimal state sequence is Q = {q1 = rainy, q2 = sunny, q3 = rainy}

Problem 2: Forward-Backward algorithm

γ choose states that are **individually** most likely. This will maximize the expected correct states at each time from 1 → T

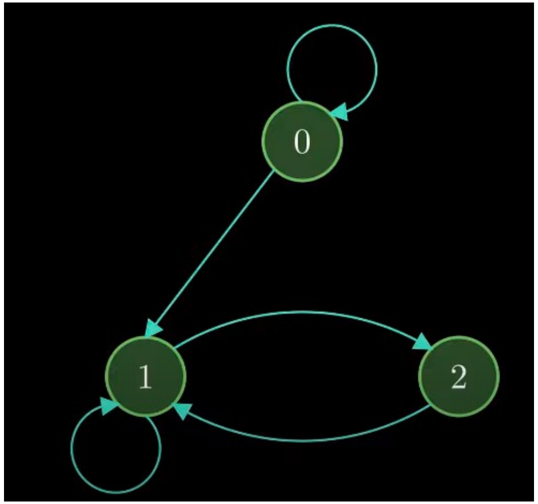
However, HMM is a model that deals with **sequential data** (the current state affects the next result).

γ reflects that we solve each step **independently**

In some cases, the solution gets stuck. Eg:

From γ we have sequence Q = {0,1,2,0}

But there's no link from 2 → 0



Problem 2: Viterbi algorithm

- Choose the path that is most likely to give the observations
- **δ_t(i)** is called joint optimal criterion at time point t
- **Ψ_i(i)** means "what state it comes from"
- Viterbi algorithm

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

    $\psi_1(i) = 0$

  - Inductive step

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

    $$\psi_t(j) = \underset{1\le i\le N}{argmax}[\delta_{t-1}(i)a_{ij}] \qquad 1 \le j \le N$$

  - Termination

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

    $$q_T^* = \underset{1\le i\le N}{argmax}[\delta_T(i)] \qquad q_t^* = \psi_{t+1}(q_{t+1}^*)$$

Path (state sequence) backtracking

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

Problem 2: Viterbi algorithm

O = H H G

initialization

$\delta_1(S) = 0.2x0.8 = 0.16$

$\delta_1(R) = 0.8x0.4 = 0.32$

$\Psi_1(S) = \Psi1(R) = 0$

inductive step

$\delta_2(S) = [\max_i(\delta_1(S)a_{SS},\ \delta_1(R)a_{RS})]b_S(H) = \max(\boxed{0.128,0.128})x0.8 = 0.1024$

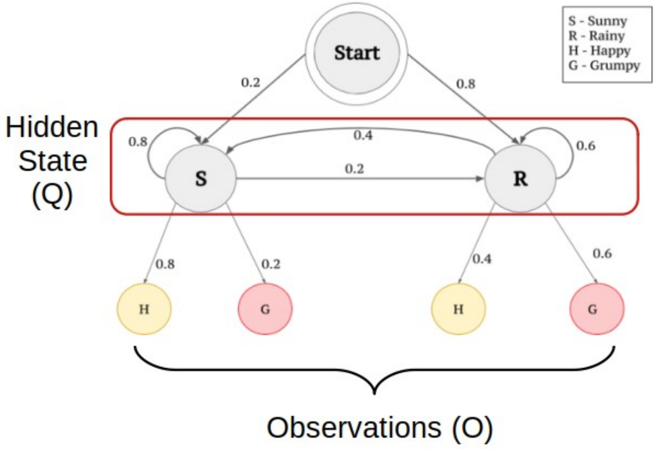
$\delta_2(R) = [\max_i(\delta_1(S)a_{SR},\ \delta_1(R)a_{RR})]b_R(H) = \max(0.032,\boxed{0.192})x0.4 = 0.0768$

$\Psi_2(S) = \mathrm{argmax}_i[\delta_1(S)a_{SS},\ \delta_1(R)a_{RS}] = \mathrm{argmax}(\boxed{0.128,0.128})$

$=>$ i= sunny/rainy

$\Psi_2(R) = \mathrm{argmax}_i[\delta_1(S)a_{SR},\ \delta_1(R)a_{RR}] = \mathrm{argmax}(0.032,\boxed{0.192})$

$= \delta1(R) =>$ i = rainy



Problem 2: Viterbi algorithm

O = H H G

- inductive step

$$\delta_3(S) = [\max_i(\delta_2(S)a_{SS}, \ \delta_2(R)a_{RS})]b_S(G) = \max(\boxed{0.082},0.031)x0.2 = 0.0164$$

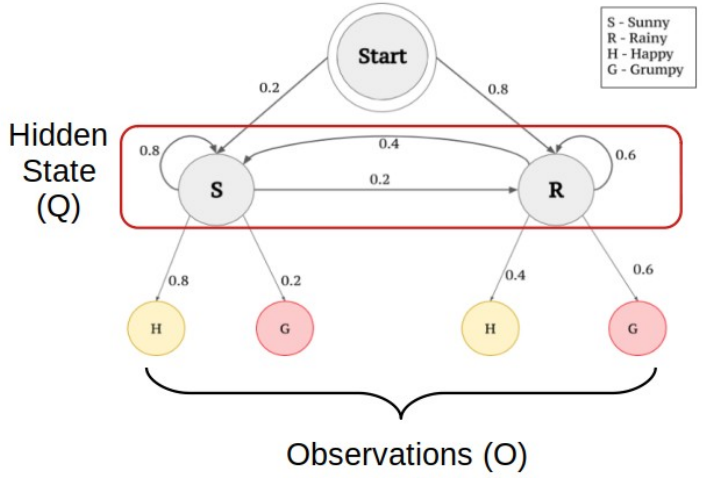
$$\delta_3(R) = [\max_i(\delta_2(S)a_{SR}, \ \delta_2(R)a_{RR})]b_R(G) = \max(0.02,\boxed{0.046})x0.6 = 0.0276$$

$$\Psi_3(S) = \operatorname{argmax}_i[\delta_2(S)a_{SS}, \ \delta_2(R)a_{RS}] = \operatorname{argmax}(0.082,0.031)$$

$$=> i= \text{sunny}$$

$$\Psi_3(R) = \operatorname{argmax}_i[\delta_2(S)a_{SR}, \ \delta_2(R)a_{RR}] = \operatorname{argmax}(0.02,0.046)$$

$$= \delta2(R) => i= \text{rainy}$$



Problem 2: Viterbi algorithm

O = H H G

Termination

According to state sequence backtracking of Viterbi algorithm

$q_3 = \underset{i}{\text{argmax}}[\delta_3(i)] = \underset{i}{\text{argmax}}[\delta_3(S), \delta_3(R)] = \underset{i}{\text{argmax}}[0.016, 0.028]$

=> i= rainy

$q_2 = \Psi_3(q_3 = R) = \Psi_3(R) = \text{rainy}$

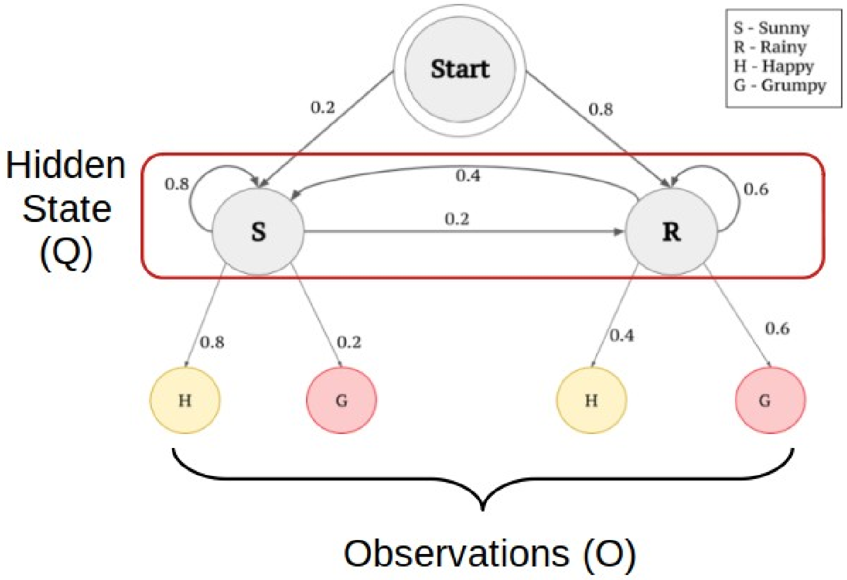
$q_1 = \Psi_2(q_2 = R) = \Psi_2(R) = \text{rainy}$

So Q = {R,R,R} most likely to give O = HHG

- Termination

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

$$q_T^* = \underset{1\le i\le N}{argmax}[\delta_T(i)] \quad q_t^* = \psi_{t+1}(q_{t+1}^*)$$



Step Ψ1

	Probability max (δ)
State = S	0.16
State = R	0.32

0.16*0.8*0.8

0.32*0.4*0.8

Step Ψ2

	Probability max (δ)
State = S	0.1024
State = R	0.0768

Step Ψ2

	Probability max (δ)
State = S	0.1024
State = R	0.0768

0.1024*0.8*0.2

0.0768*0.4*0.2

Step Ψ3

	Probability max (δ)
State = S	0.0164
State = R	0.0276

Step Ψ1

	Probability max (δ)
State = S	0.16
State = R	0.32

0.16*0.2*0.4

0.32*0.6*0.4

Step Ψ2

	Probability max (δ)
State = S	0.1024
State = R	0.0768

Step Ψ2

	Probability max (δ)
State = S	0.1024
State = R	0.0768

0.1024*0.2*0.6

0.0768*0.6*0.4

Step Ψ3

	Probability max (δ)
State = S	0.0164
State = R	0.0276

Ψ1
State = S
State = R

Ψ2
State = S
State = R

Ψ3	
State = S	0.0164
State = R	0.0276

The most likely ending state would be state = R, and the rest of the previous states could be back-traced through the arrows, which are state R at Ψ1, state R at Ψ2, and state R at Ψ3 (R-R-R).

The second likely path is R-S-S or S-S-S.

Three problems of HMM

1. Given $\lambda=(A,B,\pi)$ and a sequence of observations $O = O_1O_2...O_T$. Compute the probability that λ generated a sequence of observations, $P(O|\lambda)=?$

   Forward procedure, backward procedure

2. Given observation sequence $O = O_1O_2...O_T$ and λ. What sequence of states ($Q = q_1q_2...q_t$) best explains a sequence of observations

   Forward-backward algorithm, Viterbi

3. How to estimate $\lambda=(A,B,\pi)$ so as to maximize $P(O|\lambda)$ $\quad \lambda = (A, B, \pi)?$

   Baum-Welch (Expectation maximization)

$$(A,B,\pi) = \underset{A,B,\pi}{\operatorname{argmax}} P(O|\lambda)$$

Problem 3

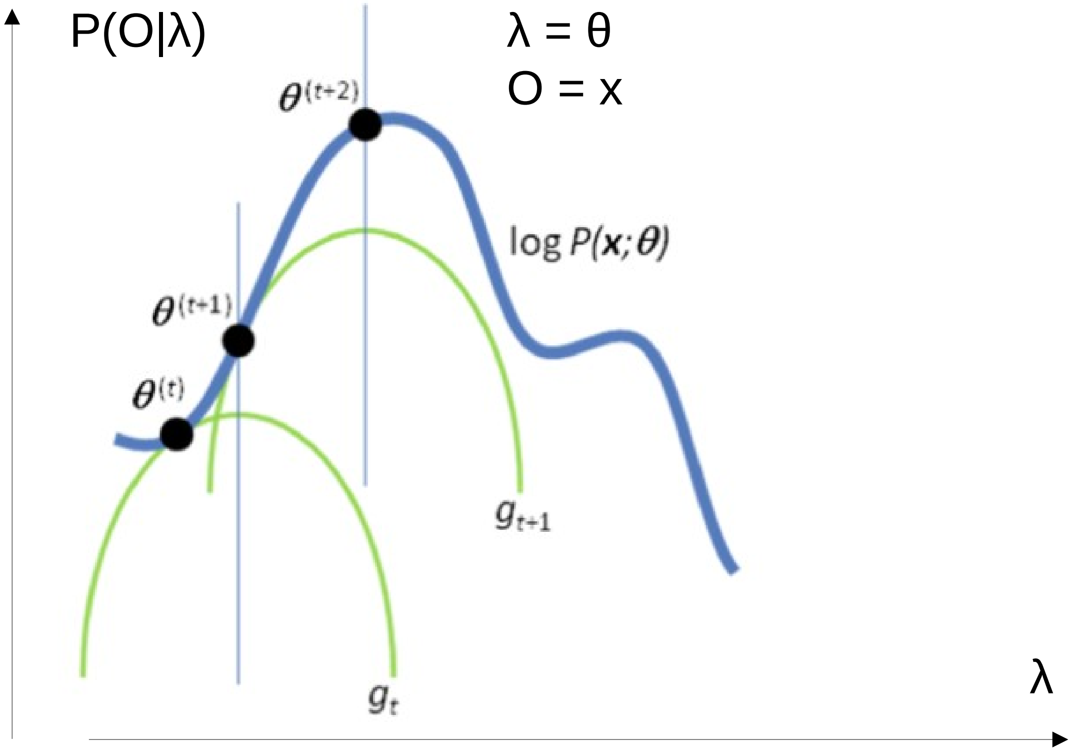
Adjust parameters such as initial state distribution π, transition probability matrix A, and observation probability matrix B so that given HMM λ gets more appropriate to an observation sequence $O = \{o_1, o_2, \ldots, o_T\}$

Note that λ is represented by these parameters (A,B,π)

$$(A,B,\pi) = \underset{A,B,\pi}{\operatorname{argmax}} \; P(O|\lambda)$$

The Expectation Maximization (EM) algorithm is applied successfully into solving problem 3, which is well-known as Baum-Welch algorithm.

The Expectation-Maximization (EM) algorithm is a general method of finding the maximum likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values.



Supplementary Figure 1 Convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function g_t that lower-bounds the objective function $\log P(x;\theta)$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of g_t. In the next E-step, a new lower-bound g_{t+1} is constructed; maximization of g_{t+1} in the next M-step gives $\theta^{(t+2)}$, etc.

Computational Statistics in Python, Duke University

EM is **iterative** algorithm that **improves parameters** after iterations **until reaching optimal parameters**.

Each iteration includes two steps: **E**(xpectation) step and **M**(aximization) step.

In E-step, the missing data are estimated given the observed data and current estimate of the model parameters.

In M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step are used in lieu of the actual missing data.

Problem 3

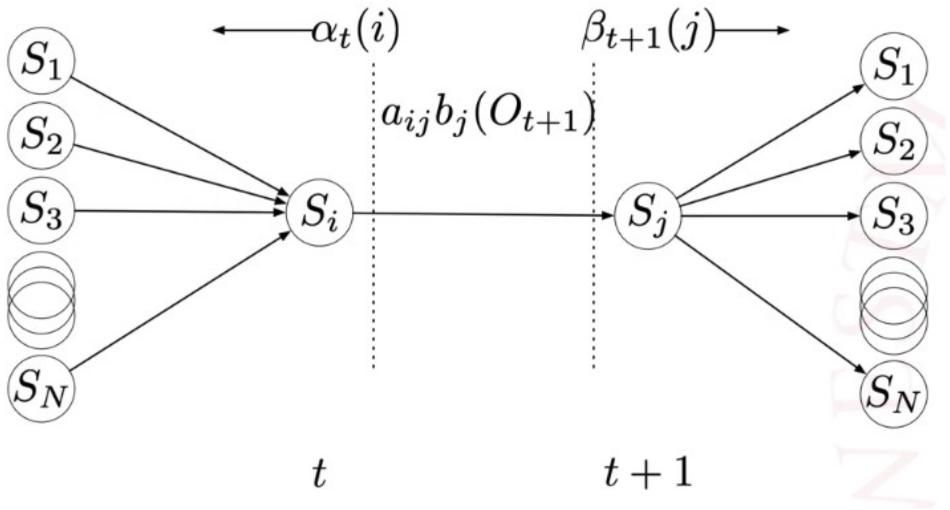
ξ_t(i,j) is the joint probability that at time t, the state is s_i and at time t+1, it is state s_j given observations O and model λ.

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

ξ_t(i,j) captures two different states

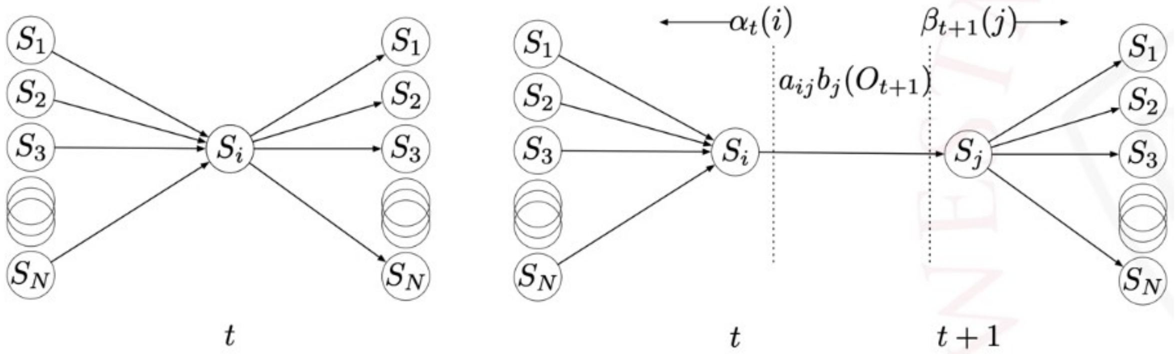
ξ_t(i,j) is constructed from forward variable and backward variable

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)}$$



$\xi_t(i,j)$ is related to $\gamma_t(i)$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$



$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from } S_i \text{ to } S_j$$

- So now…
  - We have an existing model, $\lambda = (A, B, \pi)$
  - We have a set of observations, O
  - We have a set of tools $\alpha_t(i), \beta_t(i), \gamma_t(i), \xi_t(i, j)$
- How do we use these to improve our model?

$$\bar{\lambda} = ?$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from } S_i \text{ to } S_j}{\text{expected number of transitions from } S_i}$$

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

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing } v_k}{\text{expected number of times in state j}}$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ s.t.\, O_t = v_k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\hat{\pi}_j = \frac{\gamma_1(j)}{\sum_{i=1}^{n} \gamma_1(i)}$$

$$\boxed{\begin{array}{c} \text{Given } \lambda = (A, B, \pi) \text{ and } O \text{ we can produce } \alpha_t(i), \beta_t(i), \gamma_t(i), \xi_t(i,j) \\[2em] \text{Given } \alpha_t(i), \beta_t(i), \gamma_t(i), \xi_t(i,j) \text{ we can produce } \bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi}) \end{array}}$$

Problem 3: Baum-Welch algorithm

Starting with initial value for λ (a_{ij}, $b_j(k)$, π), each iteration in EM algorithm has two steps:

1. E-step: Calculating $\xi_t(i,j)$ and $\gamma_t(i)$ given the current parameters
2. M-step: Calculating the estimate $\bar{\lambda} = (\bar{a}_{ij}, \bar{b}_j(k), \bar{\pi})$ based on $\xi_t(i,j)$ and $\gamma_t(i)$ determined at E step

The estimate $\bar{\lambda}$ becomes the current parameter for next iteration

EM algorithm stops when it meets the terminating condition, for example, the difference of current parameter λ and next parameter $\bar{\lambda}$ is insignificant (convergence).

Problem 3: Baum-Welch algorithm

O = H H G

Assume that we have initial λ as described in the table and picture

<u>At the first iteration (r=1) of E-step</u>, we have:

α_1 (S) = $\pi_S b_S$(H) = 0.16

α_1 (R) = $\pi_R b_R$(H) = 0.32

α2 (S) = (α_1(S)a_{SS} + α_1(R)a_{RS})b_S(H) = 0.2048

α_2(R) = (α_1(S)a_{SR} + α_1(R)a_{RR})b_R(H) = 0.0896

α_3(S) = (α_2(S)a_{SS} + α_2(R)a_{RS})b_S(G) = 0.039936

α_3(R) = (α_2(S)a_{SR} + α_2(R)a_{RR})b_R(G) = 0.056832

β_3 (S/R) = 1

β2 (S) = $a_{SS}b_S$(G) x 1 + $a_{SR}b_R$(G) x 1 = 0.28

β2 (R) = $a_{RS}b_S$(G) x 1 + $a_{RR}b_R$(G) x 1 = 0.44

β1 (S) = $a_{SS}b_S$(H) x 0.28 + $a_{SR}b_R$(H) x 0.44 =0.2144

β1 (R) = $a_{RS}b_S$(H) x 0.28 + $a_{RR}b_R$(H) x 0.44 = 0.1952

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O \mid \lambda)}$$

γ1 (S) = 0.354

γ1 (R) = 0.646

γ2 (S) = 0.594

γ2 (R) = 0.406

γ3 (S) = 0.412

γ3 (R) = 0.588

ξ1 (S,S) = 0.2962963

ξ1 (S,R) = 0.05820106

ξ1 (R,S) = 0.2962963

ξ1 (R,R) = 0.34920634

ξ2 (S,S) = 0.338624

ξ2 (S,R) = 0.253875

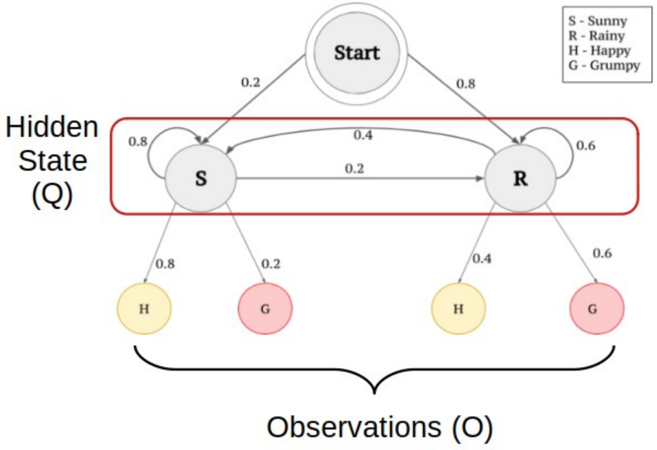
ξ2 (R,S) = 0.074074

ξ2 (R,R) = 0.333427

b_S(H) = 0.8, b_S(G) = 0.2

b_R(H) = 0.4, b_R(G) = 0.6

	S	R
π	0.2	0.8
S	0.8	0.2
R	0.4	0.6



At the first iteration (r=1) of M-step:

a_{SS} = (0.028672 + 0.032768)/(0.034304 + 0.057344) = 0.6703911

a_{SR} = (0.005632 + 0.024576)/(0.034304 + 0.057344) = 0.3296089

a_{RS} = (0.028672 + 0.007168)/(0.062464 + 0.039424) = 0.3517588

a_{RR} = (0.033792 + 0.032256)/(0.062464 + 0.039424) = 0.6482412

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

$b_S H$ = (0.034304 + 0.057344)/(0.034304 + 0.057344 + 0.039936) = 0.6964981

$b_S G$ = 0.039936/(0.034304 + 0.057344 + 0.039936) = 0.3035019

$b_R H$ = (0.062464 + 0.039424)/(0.062464 + 0.039424 + 0.056832) = 0.6419355

$b_R G$ = 0.056832/(0.062464 + 0.039424 + 0.056832) = 0.3580645

$$\bar{b}_j(k) = \frac{\sum_{t=1,\ s.t.O_t=v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

π_S = 0.354/(0.354 + 0.646) = 0.354

π_R = 0.646/(0.354 + 0.646) = 0.646

$$\hat{\pi}_j = \frac{\gamma_1(j)}{\sum_{i=1}^{n} \gamma_1(i)}$$

	S	R
π	0.3544974	0.6455026
S	0.6703911	0.3296089
R	0.3517588	0.6482412

	H	G
S	0.6964981	0.3035019
R	0.6419355	0.3580645

E-step at r=2:

$\alpha_1 (S) = \pi_S b_S(H) = 0.2469068$

$\alpha_1 (R) = \pi_R b_R(H) = 0.414371$

$\alpha 2 (S) = (\alpha_1(S)a_{SS} + \alpha_1(R)a_{RS})b_S(H) = 0.2168079$

$\alpha_2(R) = (\alpha_1(S)a_{SR} + \alpha_1(R)a_{RR})b_R(H) = 0.2246742$

$\alpha_3(S) = (\alpha_2(S)a_{SS} + \alpha_2(R)a_{RS})b_S(G) = 0.06809891$

$\alpha_3(R) = (\alpha_2(S)a_{SR} + \alpha_2(R)a_{RR})b_R(G) = 0.07773755$

$\beta_3 (S/R) = 1$

$\beta 2 (S) = a_{SS}b_S(G) \times 1 + a_{SR}b_R(G) \times 1 = 0.3214862$

$\beta 2 (R) = a_{RS}b_S(G) \times 1 + a_{RR}b_R(G) \times 1 = 0.3388716$

$\beta 1 (S) = a_{SS}b_S(H) \times 0.3214862 + a_{SR}b_R(H) \times 0.3388716 = 0.2218114$

$\beta 1 (R) = a_{RS}b_S(H) \times 0.3214862 + a_{RR}b_R(H) \times 0.3388716 = 0.2197782$

$\gamma 1 (S) = 0.21492184$

$\gamma 1 (R) = 0.78507816$

$\gamma 2 (S) = 0.42680338$

$\gamma 2 (R) = 0.57319662$

$\gamma 3 (S) = 0.45070115$

$\gamma 3 (R) = 0.54929885$

At the second iteration (r=2) of M-step:

a_{SS} = 0.64757753

a_{SR} = 0.35242247

a_{RS} = 0.34009149

a_{RR} = 0.65990851

$b_S H$ = 0.5874311

$b_S G$ = 0.4125689

$b_R H$ = 0.71204317

$b_R G$ = 0.28795683

π_S = 0.21492184

π_R = 0.78507816

At r = 25

```
In [109]: print(baum_welch(O, A, B, pi, n_iter=25))
{'A': array([[1.00000000e+00, 3.96505659e-15],
       [7.04092894e-01, 2.95907106e-01]]), 'B': array([[4.08223108e-01, 5.91776892e-01],
       [1.00000000e+00, 8.00467309e-14]]), 'gamma': array([[7.75158238e-02, 6.12310214e-01, 1.00000000e+00],
       [9.22484176e-01, 3.87689786e-01, 1.04875143e-13]])}
```

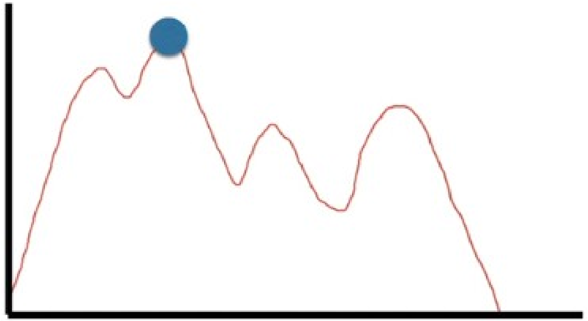
At r = 26

```
In [110]: B = np.array(((0.8, 0.2), (0.4, 0.6)))

In [111]: print(baum_welch(O, A, B, pi, n_iter=26))
{'A': array([[1.00000000e+00, 7.68485603e-16],
       [7.04103937e-01, 2.95896063e-01]]), 'B': array([[4.08232007e-01, 5.91767993e-01],
       [1.00000000e+00, 1.68209384e-14]]), 'gamma': array([[7.75192479e-02, 6.12332202e-01, 1.00000000e+00],
       [9.22480752e-01, 3.87667798e-01, 2.20379280e-14]])}
```

BAUM-WELCH ALGORITHM

- There is no known way to solve for the globally optimal parameters of lambda
- We will search for a locally optimal result
  - A result that converges to a stable good answer but isn't guaranteed to be the best answer.



References

- https://www.youtube.com/watch?v=J_y5hx_ySCg&list=PLix7MmR3doRo3NGNzrq48FItR3TDyuLCo
- https://liulab-dfci.github.io/bioinfo-combio/hmm.html
- Tutorial on Hidden Markov Model. Loc Nguyen (2016)
- https://medium.com/analytics-vidhya/viterbi-algorithm-for-prediction-with-hmm-part-3-of-the-hmm-series-6466ce2f5dc6