MA4270 Lecture 18

Markov Chain & HMM.

Graphic Models (5 lecs) $\Rightarrow$ which var influence some other vars

Consider a sequence of discrete random variables $X(1), \dots, X(t), X(t+1), \dots$

$X(t) \in S$, $|S| < \infty \Leftrightarrow S$ is finite.

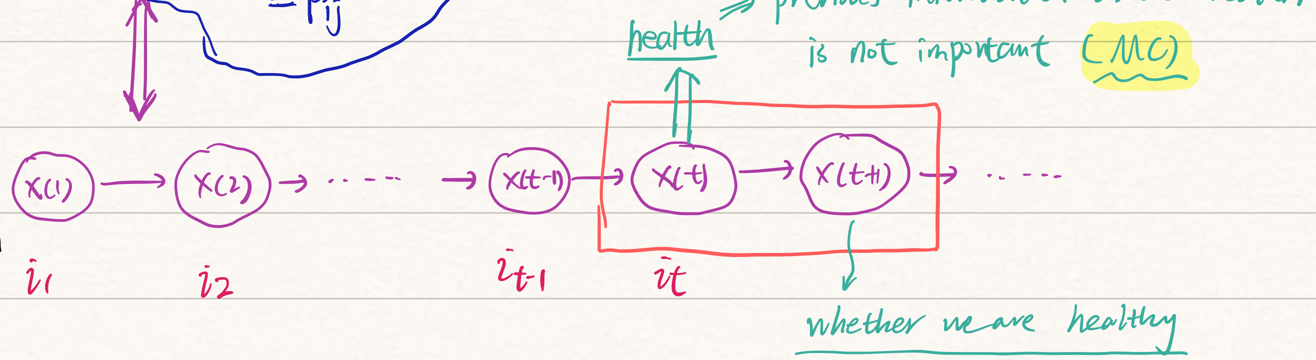
Markov Chain: $P(X(t+1) = j \mid X(t) = i) = P_{ij}$

Rmk: Prob. of transition from state i to state j depends only on i and not of the PAST!

Rmk: Homogeneous MC $\Leftrightarrow$ P_{ij} not depend on t

① $\boxed{P(X(t+1) = j \mid X(t) = i, \dots, X(1) = i_1) = P(X(t+1) = j \mid X(t) = i)} \quad \forall i, j \in S$

$= P_{ij}$



Markov (Conditional Independence) Property.

Graphical Model $\Rightarrow$ Bayesian Network / Directed GM.

Joint distribution

$P(X(1)=i_1, \ldots, X(t)=i_t)$

$= \underbrace{P(X(1)=i_1)}_{\text{Initial dist.}} \underbrace{P(X(2)=i_2 \mid X(1)=i_1) \cdots P(X(t)=i_t \mid X(t-1)=i_{t-1})}$

P_{i_{k-1}, i_k} := prob of transition from i_{k-1} to i_k

State Prediction for MC

① Claim: $\underline{P} = [P_{ij}]_{(i,j)\in S\times S} \Rightarrow$ transition Matrix

[Props]: $P(X(t+m)=j \mid X(t)=i) = [P^m]_{ij}$

Pf: $P(X(t+2)=j \mid X(t)=i)$

$= \sum_z P(X(t+2)=j, X(t+1)=z \mid X(t)=i)$

$= \sum_z P(X(t+2)=j \mid X(t+1)=z, X(t)=i)\, P(X(t+1)=z \mid X(t)=i)$

$\Downarrow$ Markov Property

$= \sum_z P(X(t+2)=j \mid X(t+1)=z)\, P(X(t+1)=z \mid X(t)=i)$

$= \sum_z P_{iz} P_{zj}$

$= [\Pi^2]_{ij}$

Initial Dist.

② Claim: $\forall n \in \mathbb{N}$ defn: $q(i) = P(X(1) = i)$ $i \in S$

$$P(X(n)=j) = \sum_{i \in S} q(i)\, P(X(n)=j \mid X(1)=i)$$

$$= \sum_{i \in S} q(i)\, [P^{n-1}]_{ij}$$

$$= [q^T P^{n-1}]_j$$

$$q(i) = \begin{bmatrix} q(1) \\ \vdots \\ q(k) \end{bmatrix}_{k \times 1}$$

Suppose that $S = \{1, 2, \dots, k\}$

That is, the j-th comp. of $q^T P^{n-1}$ is just $P(X(n)=j)$

Sum over all intermediate states until $X(n) = j$

$$\sum_{\boxed{i_1, \dots, i_n}} P(X(1)=i_1) \prod_{t=1}^{n-1} P(X(t+1)=i_{t+1} \mid X(t)=i_t)$$

k^n

$$= q^T \cdot \underbrace{P \cdots P}_{n-1} \mathbb{1}_{k \times 1}$$

$= 1.$

$O(k^n)$ ⇒ to heavy

⇒ Summing over k^n possible state configurations, which can be done very efficently using MATRIX PRODUCTS

dynamic Programming

Forward message: (Passing)

$\alpha_t(i) = P(X(t) = i)$ can be done recursively!

⇓

$$\alpha_t^T = q^T \underbrace{P \cdots\cdots P}_{t-1}$$

Recursive

① $q^T = \alpha_1^T$
$$= [Pr(X(1)=1, \ldots, Pr(X(1)=k)]$$

② $\alpha_2^T = \alpha_1^T P$

⋮

(t) $\alpha_t^T = \alpha_{t-1}^T P$

$\Updownarrow$

$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) P_{ij}$$

If I want to compute $P(X(n)=j)$, this takes time $O(k^2 n)$

Estimation of paras. (q, P)

Q: Given observed $x_1, \ldots, x_n \in S$, can we estimate q & P ?

Answer: MLE

Log-likelihood

$\downarrow$

$$\log P(X) = \log \left[P(X(1) = x_1 \prod_{t=1}^{n-1} P(X(t+1) = x_{t+1} \mid X(t) = x_t) \right]$$

$$= \log q(x_1) + \sum_{t=1}^{n-1} \log P_{x_t, x_{t+1}}$$

$$= \log q(X_1) + \sum_{(i,j)\in S^2} \hat{n}(i,j) \log P_{ij}$$

where $\hat{n}(i,j) = \sum_{t=1}^{n-1} \mathbb{1}\{X_t = i, X_{t+1} = j\}$

the # of times the MC transition from i to j

Use MLE Principle to maximize the log-likelihood over q & $\underline{P}$.

① $\underline{P}$.

$\underline{P}$: Stochastic matrix $\rightarrow$ $\underbrace{\forall i, \ \sum_j P_{ij} = 1}$

k equality constraint

If we maximize the log-likelihood w.r.t P and take equality constraints into account, $\Rightarrow$ Lagrangian Multiplier Application

$$\Downarrow$$

$$\hat{P}_{ij} = \frac{\hat{n}(i,j)}{\sum_{j'} \hat{n}(i,j')} \quad \forall i,j \in S$$

check : $\sum_j \hat{P}_{ij} = 1$.

If we adopt a Bayesian approach, then we would put a Dirichlet prior on each $P_i = (P_{i1}, \dots, P_{ij})$

$$\hat{P}_{ij} = \frac{\hat{n}(i,j) + 1}{\sum_{j'} (\hat{n}(i,j') + 1)}$$

Prior Belief

Hidden Markov Models (HMM)

- Forward-Back Alg.
- Viterbi Alg.
- B-W (EM Alg.)

Background: Biological Sequence Analysis

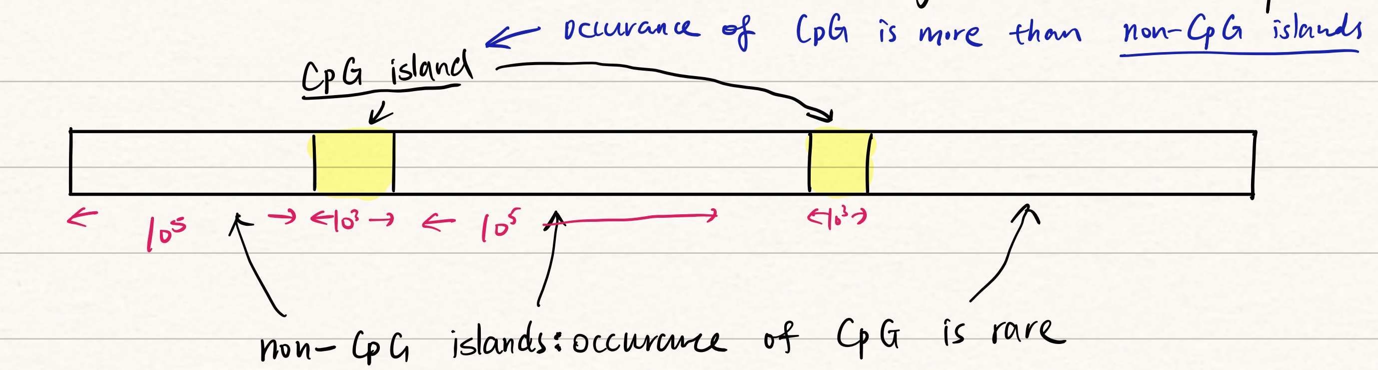
DNA code consists of 4 chemical bases: A T C G

DNA bases are paired together { A – T, C – G } to form

base pairs attached to the Sugar-Phosphate Backbone

Pattern: in which 4 bases occur in DNA is not unif. at random

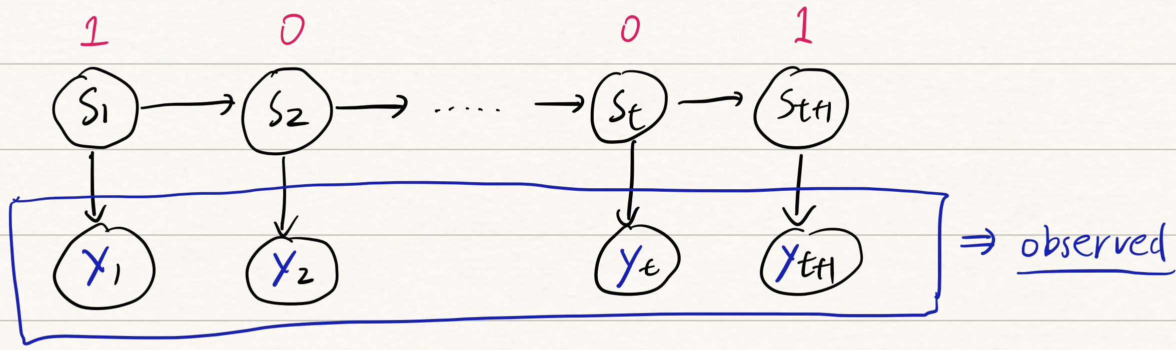
→ CG (CpG) occur rarer in the gene than expected



Q1: Given a short sequence, is it from a CpG island?

Q2: Given a long sequence, does it contain a CpG island?

$S = \{ \underbrace{\text{CpG Island}}_{1} , \underbrace{\text{Non-CpG Island}}_{0} \}$

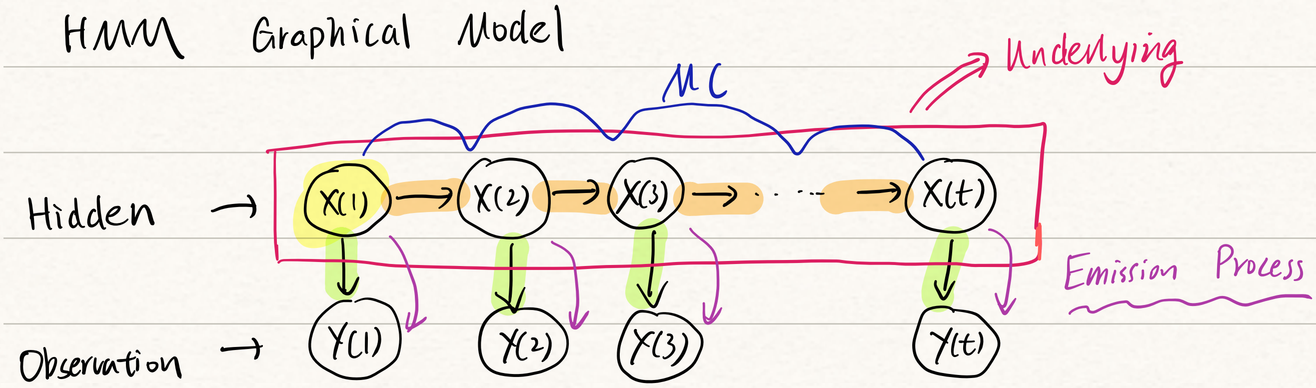


$Y_i \in \{ aa, ac, ag, \dots, cg, \dots, tt \}$

$\downarrow$

$\begin{cases} a \to a \\ a \to c \\ \vdots \\ t \to t \end{cases}$

HMM Graphical Model



$$P(x_1, \dots, x_n, y_1, \dots, y_n) = P(x_1)\, P(y_1 | x_1) \prod_{t=1}^{n-1} \underbrace{P(x_{t+1} | x_t)}_{\text{transition}} \underbrace{P(y_{t+1} | x_{t+1})}_{\text{Emission}}$$

Initial State Dist^{n}.

Three Problems To Solve

① Evaluate the Probability of Data

$$P(y_1, \dots, y_n) = \sum_{x_1, \dots, x_n} P(x_1, \dots, x_n, y_1, \dots, y_n)$$

same thing

{ Belief Propagation Alg.
{ α-β Algorithm

Forward-Backward Alg. can solve this in time $O(n)$

If we do this naively,
it will $\underbrace{\text{take } O(k^n) \text{ time}}_{\text{summation}}$

② Most likely hidden/latent state sequence ⇒ suppose we know
(Inference) p, q, $P(y|x)$

$$(x_1^*, \ldots, x_n^*) = \arg\max_{x_1, \ldots, x_n} P(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Viterbi Algorithm

③ Estimation paras of HMM

Baum-Welch Algorithm (EM).

Problem 1: Forward-Backward Alg. $x_1 \{k \quad \cdots \quad x_n \{k$

$$P(y_1, \ldots, y_n) = \overbrace{\sum_{x_1, \ldots, x_n}}^{一共k^n个} P(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

(Naive computation takes $\overset{\text{summation}}{O(k^n)}$ time. Infeasible!)

Similar to Forward Inference for MC.

Only difference is that now we have $\boxed{\text{Emission Prob.}}$ $P(y|x)$

$$\forall y. \rightarrow D_y = \begin{bmatrix} P(y|1) & & \\ & \ddots & \\ & & P(y|k) \end{bmatrix} \in \mathbb{R}^{k \times k} \quad \text{(Diagonal Matrix)}$$

D_y is constructed from Emission Matrix

Now we have q, p, Emission Matrix

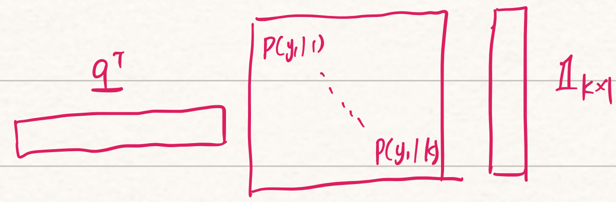
Dy

TRY SIMPLE CASE FIRST

$$P(y_1) = \sum_i P(X_1 = i)\, P(\cancel{Y_1} = y_1 \mid X_1 = i)$$ ⇒ no transition matrix

$$= \sum_i q(i)\, P(y_1 \mid i)$$

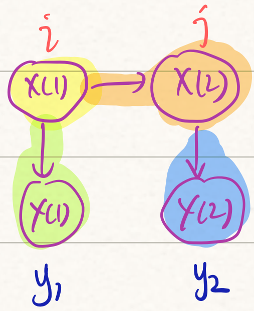
$$= q^T P_{y_1} \mathbb{1}_{k\times 1}$$



HARDER ONE

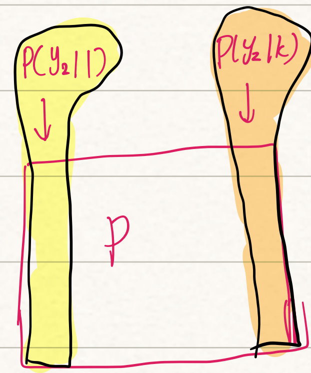
$$\sum_{ij} q(i)\, P(y_1 \mid i)\, P_{ij}\, P(y_2 \mid j)$$

$$P(y_1, y_2) = \sum_i q(i)\, P(y_1 \mid i) \sum_j P_{ij}\, P(y_2 \mid j)$$



$$= q^T D_{y_1} P D_{y_2} \mathbb{1}$$

$P D_{y_2} =$



解释

$[\,q(1) P(y_1 \mid 1), \dots, q(k) P(y_1 \mid k)\,]$

$$\begin{bmatrix} \sum_j P_{1j} P(y_2 \mid j) \\ \vdots \\ \sum_j P_{kj} P(y_2 \mid j) \end{bmatrix}$$

→ 代表 $1 \to j \to$ for all possible j

$y_1 \quad y_2$

Conclusion:

$$P(y_1, \dots, y_n) = \overbrace{\underbrace{q^T D_{y_1}}_{\alpha_1^T} P D_{y_2}}^{\alpha_2^T} \underbrace{\cdots P D_{y_n} \mathbb{1}}_{\beta_3} \Rightarrow O(k^2 n)$$

computational complexity

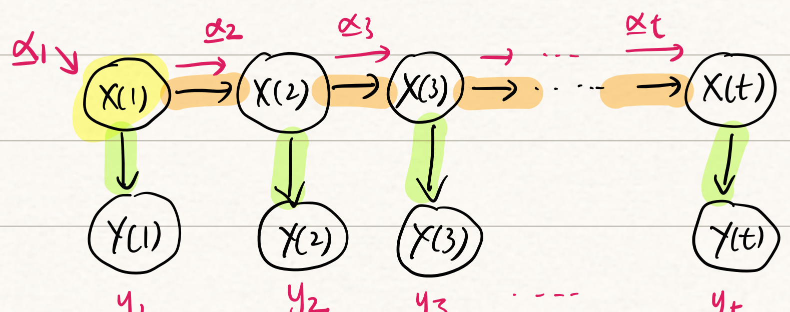
Therefore: define

$$\alpha_1^T = q^T D_{y_1}$$

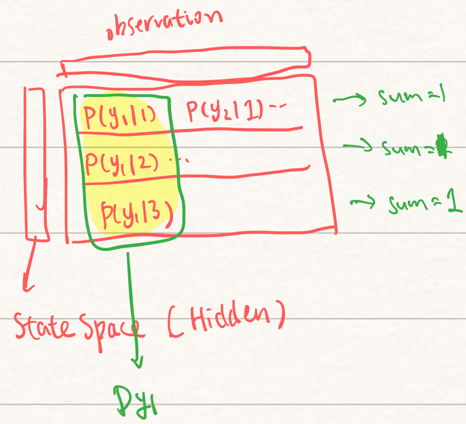
$$\alpha_2^T = \alpha_1^T P D_{y_2} \quad \Rightarrow \quad P(y_1, \dots, y_n) = \alpha_n^T \mathbb{1}.$$

$$\vdots$$

$$\alpha_t^T = \alpha_{t-1}^T P y_t$$



① q ② P ③ D



observation
P(y1|1) P(y2|1) --
P(y1|2) ...
P(y1|3)
sum=1
sum=
sum=1
State Space (Hidden)
Py1