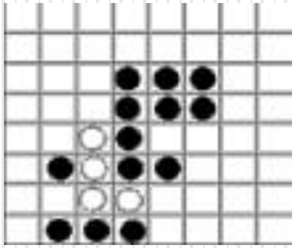
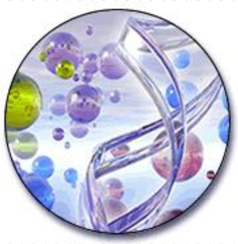
Welcome to

Introduction to Machine Learning!

2010.8.5





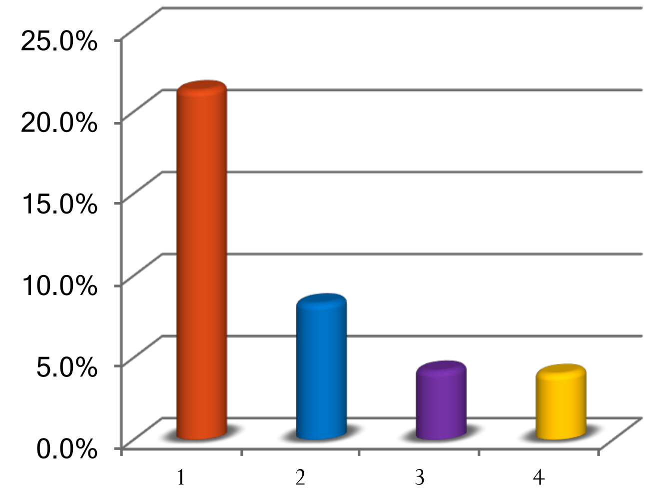
Coffee Time

Min Zhang

z-m@tsinghua.edu.cn

Email Spam Filtering

- **Volume of spam:** 90 billion/day, 80% by <200 senders
- The major sources of spam in the fourth quarter (by Sophos)
  - The United States (the origin of 21.3% of spam messages, down from 28.4%)
  - Russia (8.3%, up from 4.4%)
  - China (4.2%, down from 4.9%)
  - Brazil (4.0%, up 3.7%)



Lessons learned from Email-spam filtering

- Don't arbitrarily ignore any data
  - E.g. email header
- Different Cost: **False positive** v.s. **False negative**
- A very good talk: http://www.paulgraham.com/better.html

Email spam filtering (cont.) (By the talk)

- The first papers about Bayesian spam filtering seem to have been two given at the same conference in 1998
  - 1) by Pantel and Lin; 2) by a group from Microsoft Research.
- Pantel and Lin's filter was the more effective of the two
  - **But it only caught 92% of spam, with 1.16% false positives.**
- When I tried writing a Bayesian spam filter
  - **It caught 99.5% of spam with less than .03% false positives**
- 5 differences

Subject*FREE 0.9999

Subject*free 0.9782,

free 0.6546

free!! 0.9999

1. They trained their filter **on very little data**: 160 spam and 466 nonspam mails.
2. The most important difference is probably that **they ignored message headers**.
3. Pantel and Lin **stemmed the tokens** -- this is a kind of premature optimization
4. They calculated probabilities differently. **They used all the tokens, whereas I only use the 15 most significant**.
5. They **didn't bias against false positives**. I do this by **counting the occurrences of tokens in the nonspam corpus double.**

Topic 3 (cont.) Bayesian and MDL

贝叶斯学习与最小描述长度原则

MDL (Minimum Description Length)

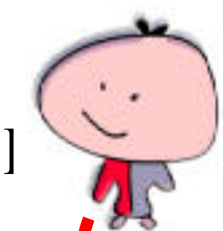
- Occam's razor:
  - prefer the shortest hypothesis
- MDL:
  - prefer the hypothesis *h* that minimizes:

$$h_{\mathsf{MDL}} = \underset{h \in \mathcal{H}}{argmin} \{ L_{C_1}(h) + L_{C_2}(D|h) \}$$

where $L_C(x)$ is the description length of x under encoding C

Explanation to MDL（information theory）

- Code design for randomly send messages
  - the probability to message i is p_i
- What's the optimal (shortest expected coding length) code?
  - Assign shorter codes to messages that are more probable
  - The optimal code for message i is $-\log_2 p$ bits [Shannon & Weaver 1949]



- $-\log_2 p\,(h)$:   length of h under optimal code C
- $-\log_2 p\,(D\,|\,h)$: length of D given h under optimal code C

MDL and MAP

$$\begin{aligned} h_{\mathrm{MAP}} &= \underset{h \in \mathcal{H}}{argmax}\, \mathrm{P}(D|h)\mathrm{P}(h) \\ &= \underset{h \in \mathcal{H}}{argmax}\{\log_2 \mathrm{P}(D|h) + \log_2 \mathrm{P}(h)\} \\ &= \underset{h \in \mathcal{H}}{argmin}\{\underbrace{-\log_2 \mathrm{P}(D|h)}_{L_{C2}(D|h)} \underbrace{-\log_2 \mathrm{P}(h)}_{L_{C1}(h)}\} \\ &= h_{\mathrm{MDL}} \end{aligned}$$

Another Explanation to MDL

$$h_{\mathrm{MDL}} = \underset{h \in \mathcal{H}}{argmin} \{ L_{C_1}(h) + L_{C_2}(D|h) \}$$

- length of h, and the cost of encoding data given h
  - Suppose the sequence of instances is already known to both transmitter and receiver
  - No misclassification: no need to transmit any information given h
  - Some are misclassified by h: need transmit
  - 1. which example is wrong?

    -- at most $\log_2 m$ (m: # of instances)
  - 2. the correct classification?

    -- at most $\log_2 k$ (k: # of classes)

Explanation to MDL

$$h_{\text{MDL}} = \underset{h \in \mathcal{H}}{argmin}\{L_{C_1}(h) + L_{C_2}(D|h)\}$$

- Tradeoff: complexity of hypothesis vs. the number of errors committed by the hypothesis
  - Prefer a shorter hypothesis that makes a few errors

    Not a longer hypothesis that perfectly classifies the training data

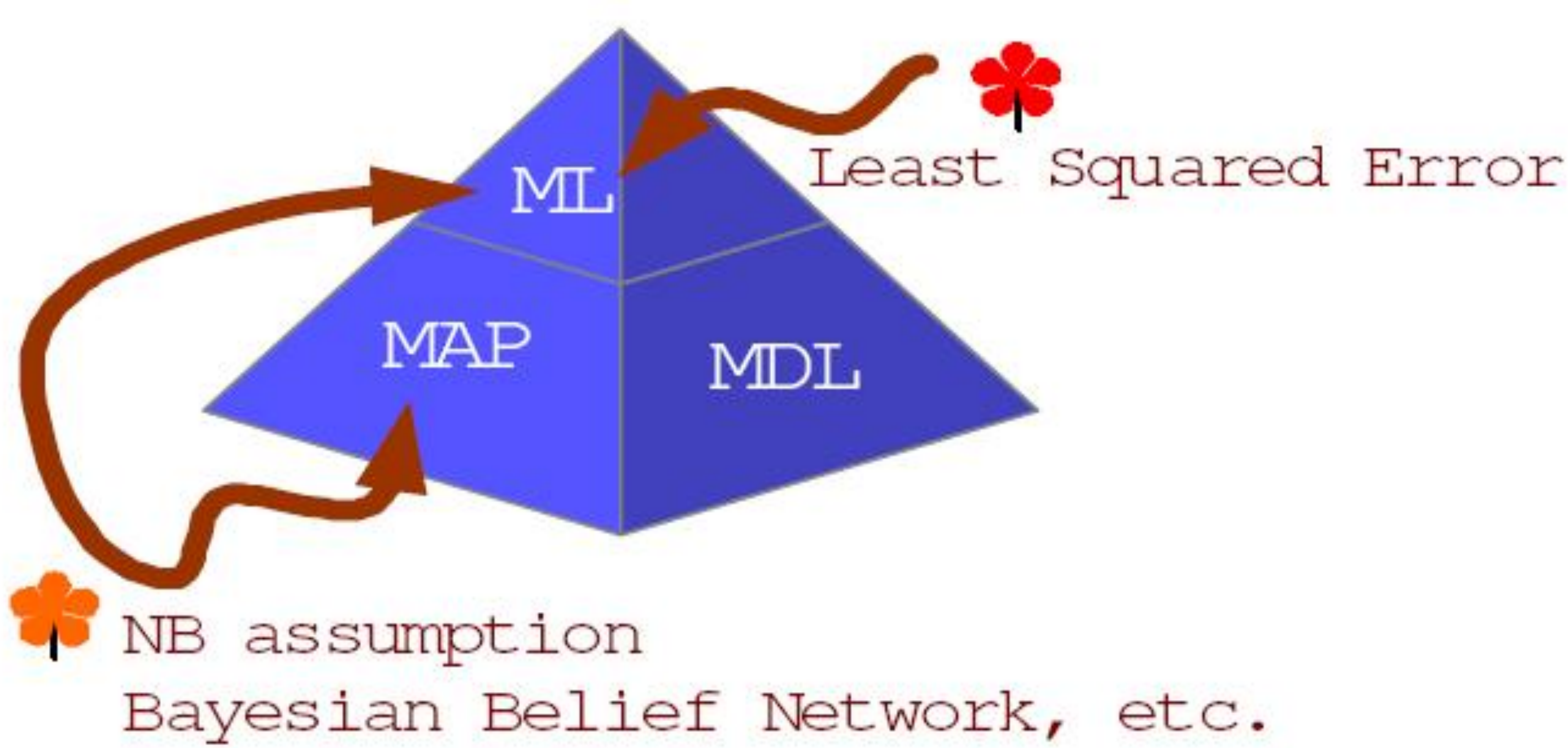
dealing with ***overfitting*** problem

Overview

- Bayes theorem
  - Use prior probability to inference posterior probability
- Max A Posterior, MAP, h_{MAP} (极大后验假设)
- Maximum Likelihood, ML, h_{ML} (极大似然假设)
  - ML vs. LSE (Least Square Error)
- Naïve Bayes, NB, 朴素贝叶斯
  - Independent assumption
  - NB vs. MAP
- Maximum description length, MDL (最小描述长度)
  - Tradeoff: hypothesis complexity vs. errors by h
  - MDL vs. MAP

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

Overview: MAP_MDL_ML_NB

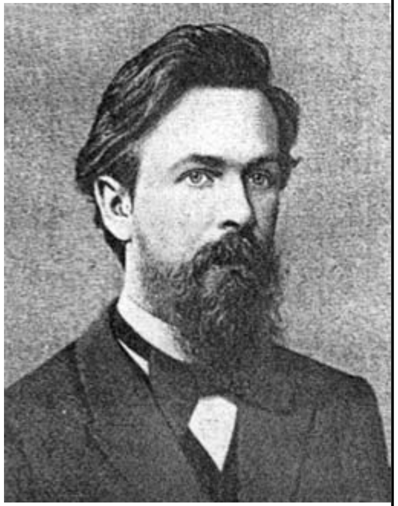


Topic 4. Markov Model

马尔可夫模型

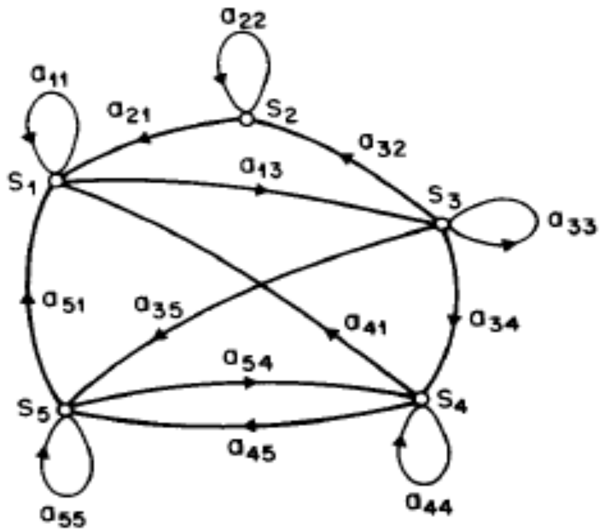
Andrey A. Markov (马尔可夫)

- Russian mathematician (1856 ~ 1922)
- His work was very highly esteemed by Chebyshev, " represents one of the finest achievements of the St Petersburg school of number theory,"
- 1896, ordinary academician of Russian Academy of Sciences
- Contributions
  - Number theory, Probability theory,
  - Especial remarkable research on Law of Large Numbers, the central limit theorem
- Proposed Markov Chain in 1907
  - This work founded a completely new branch of probability theory and launched the theory of stochastic processes.



MM & HMM

- Application background
  - Weather prediction： rain, sunny, cloudy…
  - Predict the infection of communicable diseases
  - Speech recognition
  - Chinese Input Method
  - Population prediction
  - Bioinformatics (e.g. gene analyses)
  - ……



Markov Model

Example: weather prediction problem

- Three types of weather: {*sunny*, *rainy, foggy*}
- The weather today is based on what the weather was like yesterday, the day before, …

  $P(w_n | w_{n-1}, w_{n-2}, \ldots, w_1)$
- If we know in the past 3 days (in order): sunny, sunny, foggy
- Then the probability that tomorrow would be rainy is:

$$P(w_4 = Rainy \mid w_3 = Foggy, w_2 = Sunny, w_1 = Sunny)$$

Markov Process

Markov Model

- Given a sequence of data $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \ldots, \mathbf{x}_N\}$
- 1st order Markov assumption (一阶马尔可夫假设)
  - The probability of an observation at time *n* only depends on the observation at time *n*-1.

$$p(\mathbf{x}_n | \mathbf{x}_{n-1}, \ldots, \mathbf{x}_2, \mathbf{x}_1) \approx p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

$$p(\mathbf{D}|\mathcal{M}) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

- **The transition probabilities (转移概率) $a_{i,j}$** —— given the state S_i at time *n-1*, the probabilities of being in state S_j at time *n*:

$$a_{i,j} = p(x_n = S_j | x_{n-1} = S_i)$$

Properties of Transition Probabilities

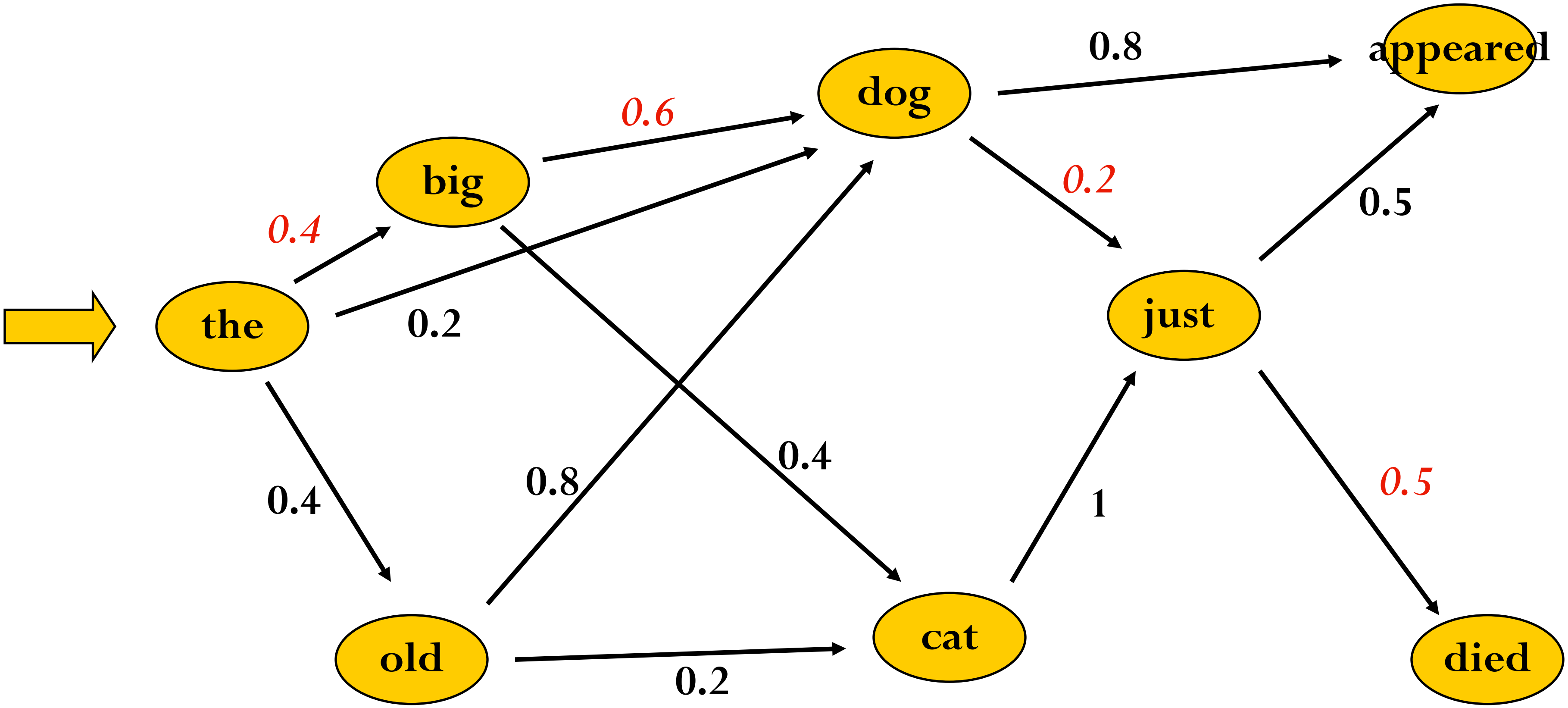
- $a_{i,j} \geq 0$
- Time invariant (homogeneous 时间不变性)

$$p(x_n = S_j | x_{n-1} = S_i) = p(x_{n+T} = S_j | x_{n-1+T} = S_i)$$

- Transition matrix： row sum = 1

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \sum_{j=1}^{M} a_{i,j} = 1 \text{ for } i = 1..M$$

Graphic illustration of a Markov Model -- an example

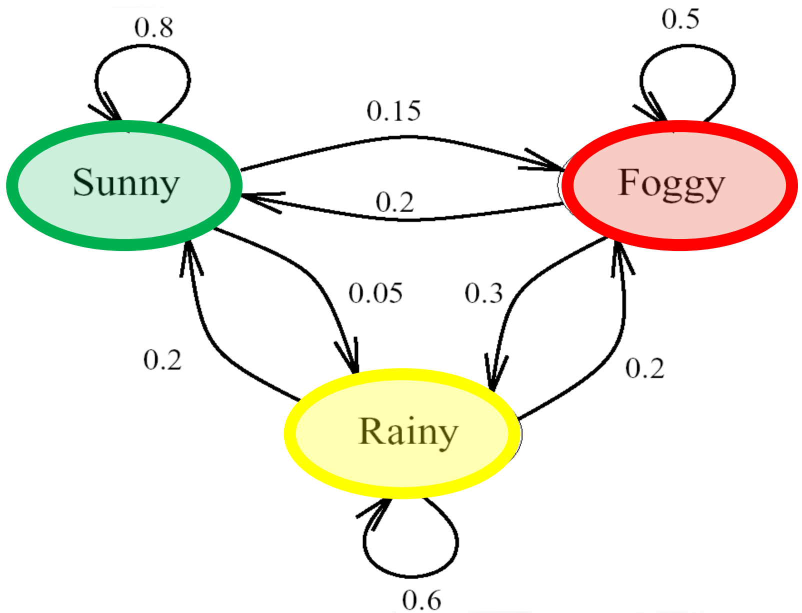


P("the big dog just died") = 0.4 * 0.6 * 0.2 * 0.5

Back to the weather prediction

$$P(w_{\text{tomorrow}} \mid w_{\text{today}})$$

		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5



- Using the Markov Assumption :

$$P(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_1) \approx P(w_n \mid w_{n-1})$$

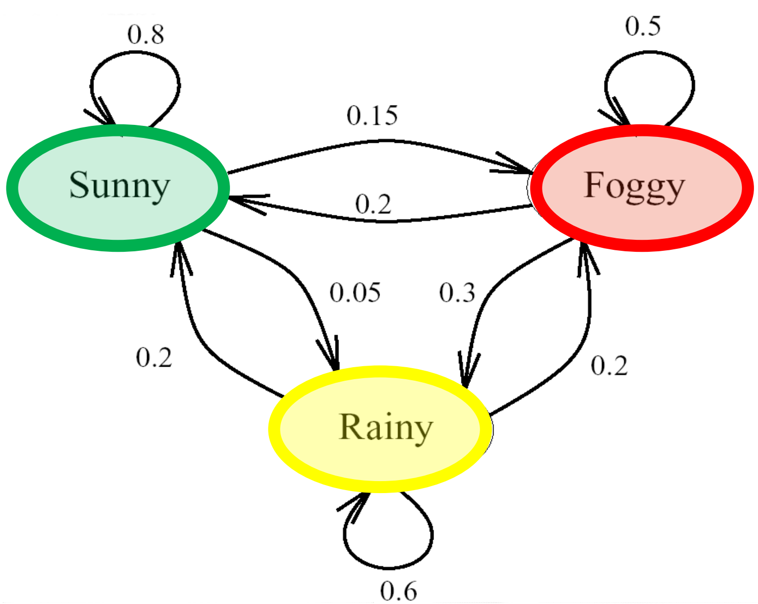
- And the joint probability…

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1})$$

Question 1

- Given: today – sunny,
- What's the probability of tomorrow – sunny & the day after – rainy?

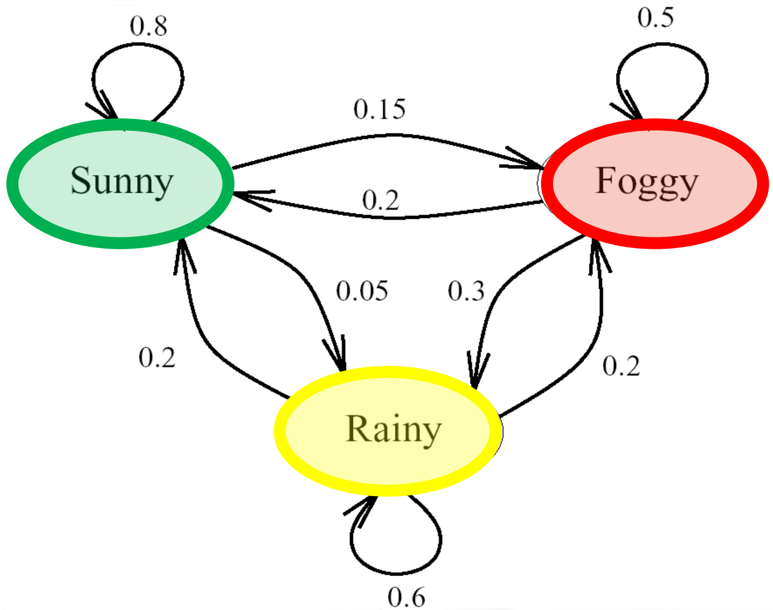
$$P(w2 = \text{Sunny}, w3 = \text{Rainy} \mid w1 = \text{Sunny}) = ?$$



Question 2

- Given: today – foggy
- What's the probability that two days from now – rainy?

$$P(w3 = \text{Rainy} \mid w1 = \text{Foggy}) = ?$$



Topic 5: Hidden Markov Model

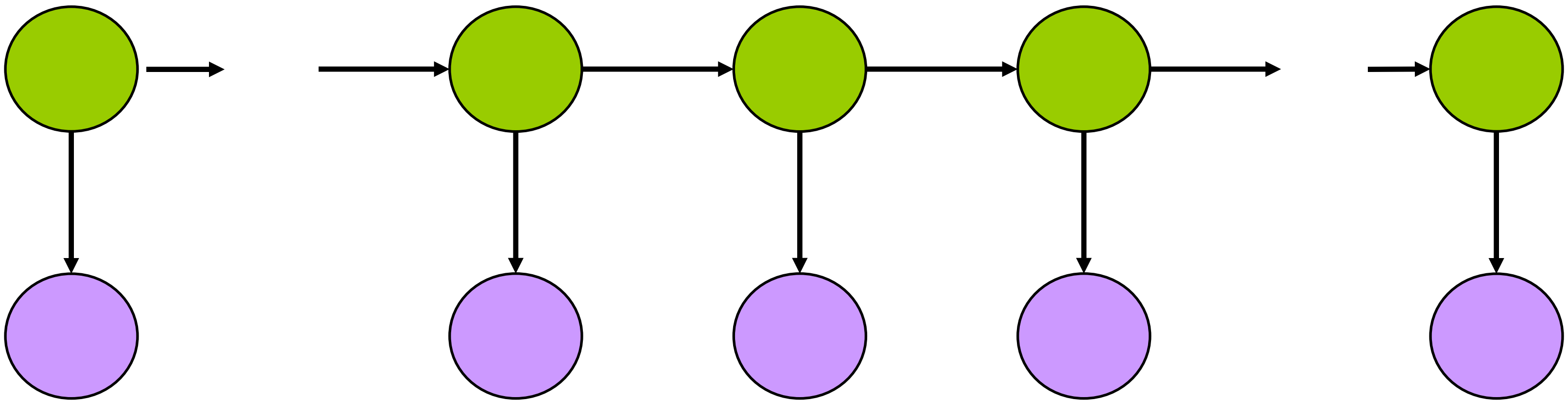
隐马尔可夫模型

Hidden Markov Model

- What is a Hidden Markov Model (HMM)?
- Weather prediction problem
  - Suppose one was locked in a room for several days,
  - And he wanted to know the weather outside.
  - The only piece of evidence he had: whether the caretaker (person who brings food to him) brought an umbrella or not.

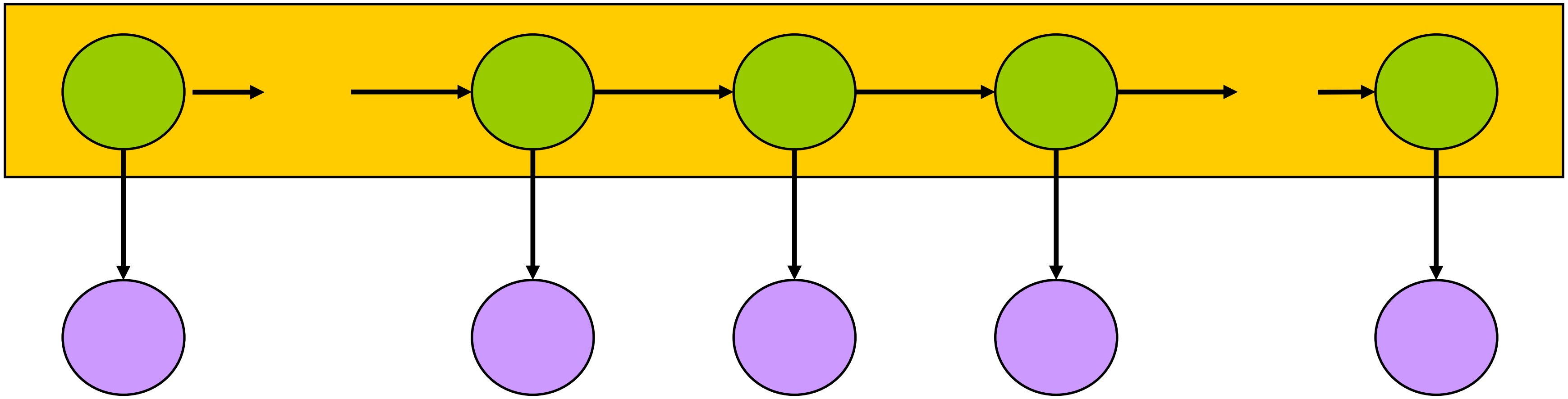
Observed state ： umbrella?	Hidden state: the weather

What is an HMM?



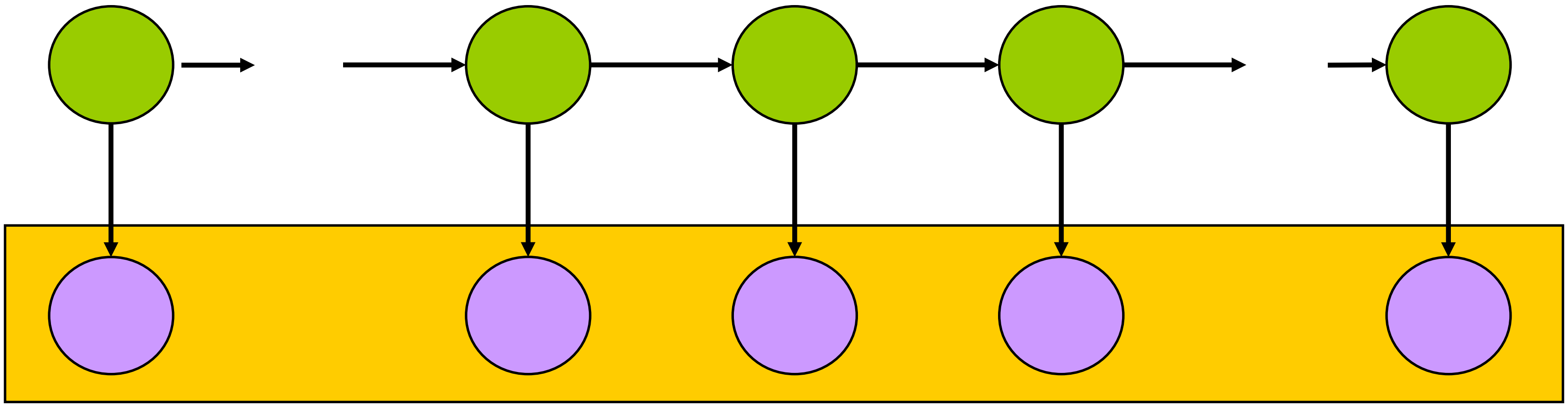
- Graphical Model
- Circles: States
- Arrows: Probabilistic dependencies between states

What is an HMM?



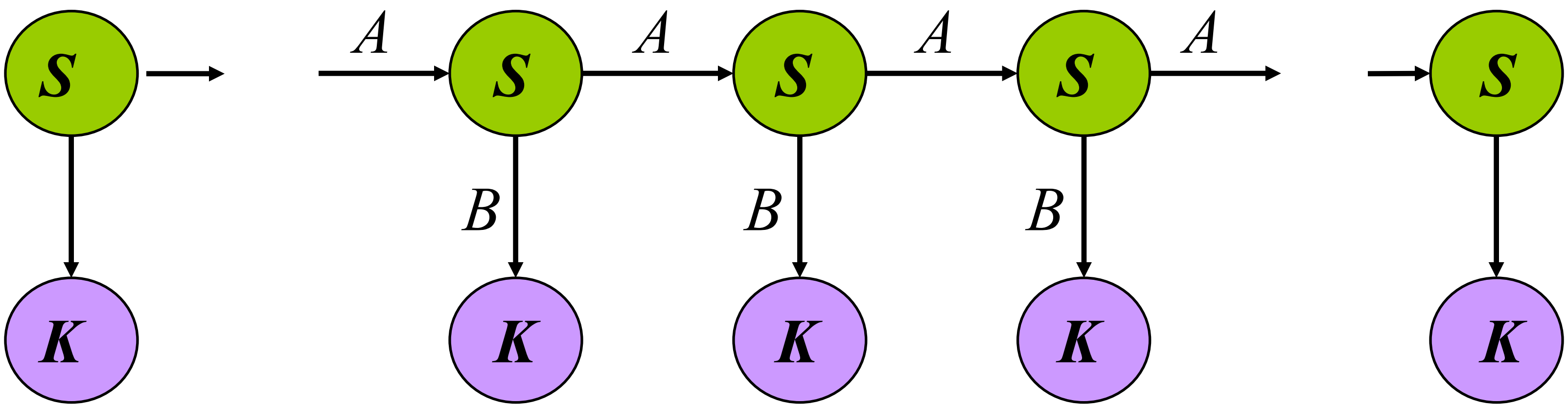
- Green circles are *hidden states*
- Dependent only on the previous state
- "The past is independent of the future given the present."

What is an HMM?



- Purple nodes are *observed states*
- Dependent only on their corresponding hidden state

HMM formalism



- $\{S, K, \Pi, A, B\}$
- S : $\{s_1 \ldots s_N\}$ values for the hidden states
- K : $\{k_1 \ldots k_M\}$ values for the observations
- $\Pi = \{\pi_i\}$ the initial state probabilities
- $A = \{a_{ij}\}$ the state transition probabilities $p(x_n{=}S_j|x_{n-1}{=}S_i){=}a_{i,j}$
- $B = \{b_{ij}\}$ the observation state probabilities (输出概率) $p(y_n{=}K_j|x_n{=}S_i){=}b_{i,j}$

HMM for Weather Prediction

- Recap: the weather Markov process

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1})$$

- Now, the actual weather is hidden, we can only observe whether the caretaker brings an umbrella.

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n | w_1, \ldots, w_n) P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)}$$

Where u_i is true if the caretaker brought an umbrella on day i, and false if he didn't.

HMM for Weather Prediction (cont.)

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n | w_1, \ldots, w_n) P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)}$$

- Assume that, for all i, given w_i, u_i is independent of all u_j and w_j for all $j \neq i$

  -- independent observation assumption （输出独立性假设）

$$P(u_1, \ldots, u_n | w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(u_i | w_i)$$

	With umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Question 3

- The observable variable can take two values
  - {C1 = umbrella, C2 = no umbrella}.
- The hidden variable consists of three states
  - $\{S_1 = \text{sunny}, S_2 = \text{rainy}, S_3 = \text{foggy}\}$
- Suppose that the day one was locked into the room is sunny.
- The second day the caretaker carried an umbrella into the room
- Now what is the probability that the second day was rainy?

$$p(x_2 = S_2 | y_2 = C_1, x_1 = S_1)$$

HMM: 3 basic problems

Given initialized HMM $\mu = \{A, B, \pi\}$

The observation sequence $\sigma = O_1, \ldots, O_T$

- Estimation problem: Compute the probability of a given observation sequence $p(\sigma \mid \mu)$
- Decoding problem: Given an observation sequence, compute the most likely hidden state sequence
- Learning problem: Given an observation sequence and set of possible models, which model most closely fits the data

HMM: 3 basic problems

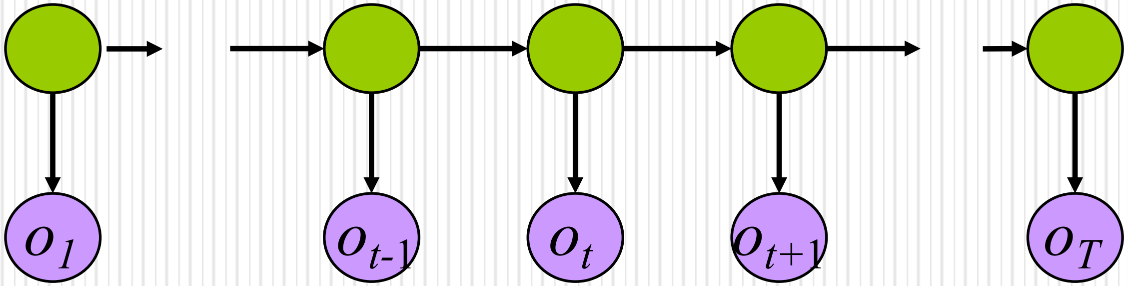
Given initialized HMM $\mu = \{A,B,\pi\}$

The observation sequence $\sigma = O_1,\ldots,O_T$

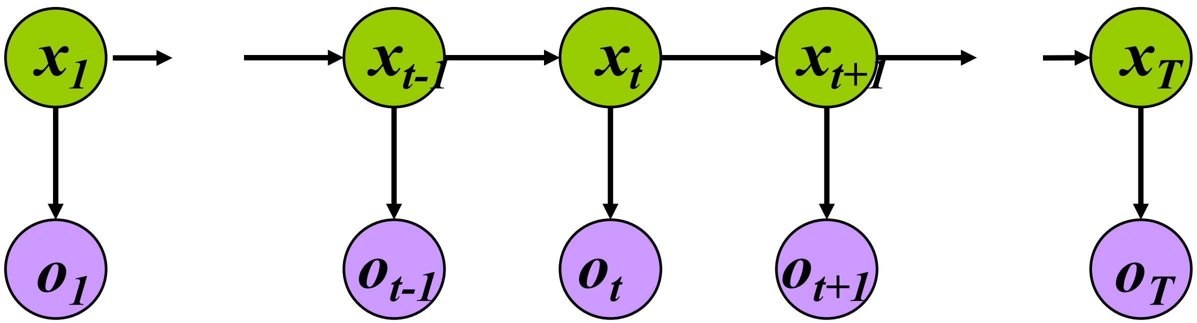
- Estimation problem: Compute the probability of a given observation sequence $p(\sigma \mid \mu)$
- Decoding problem: Given an observation sequence, compute the most likely hidden state sequence
- Learning problem: Given an observation sequence and set of possible models, which model most closely fits the data

Solutions to Problem1 -- Estimation prob.

$$O = (o_1 \ldots o_T),\ \mu = (A, B, \Pi),\quad \text{Compute } P(O \mid \mu)$$



Estimation problem – Solution 1

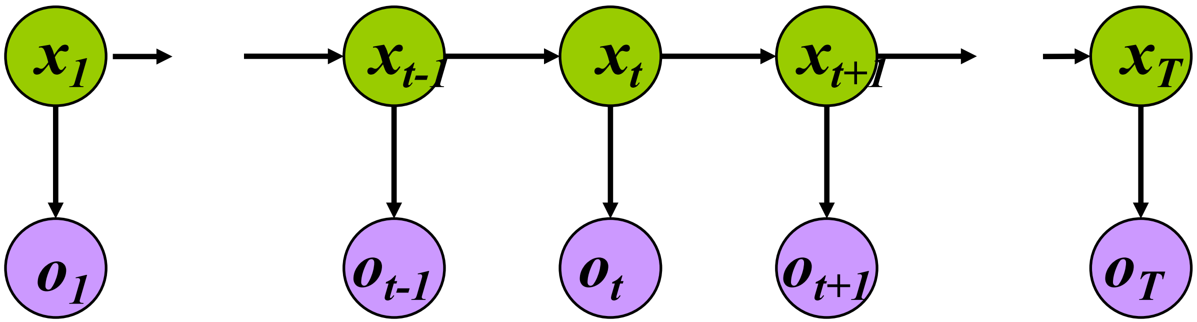


$$P(O \mid \mu) = ?$$

$$\boxed{P(O \mid \mu) = \sum_X P(O, X \mid \mu)}$$

$$P(O, X \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)$$

Estimation problem – Solution 1



$$P(O \mid \mu) = ?$$

$$P(O \mid \mu) = \sum_X P(O, X \mid \mu)$$

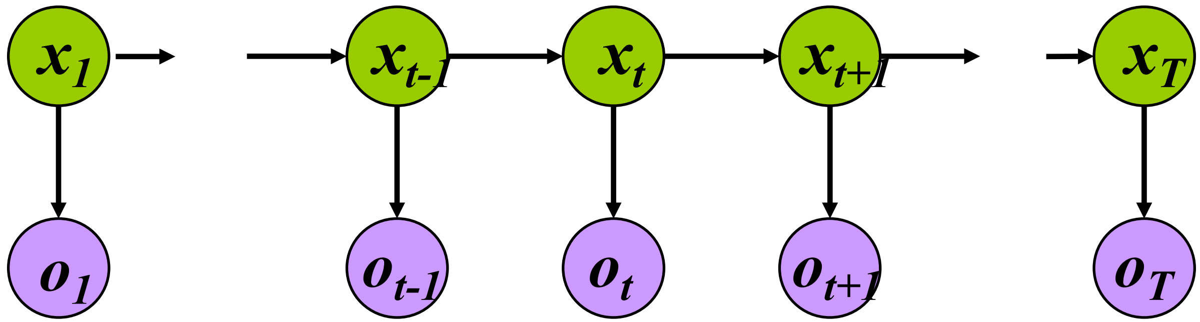
$$\boxed{P(O \mid \mu) = \sum_X P(O \mid X, \mu) P(X \mid \mu)}$$

$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

$$\boxed{P(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}}$$

Estimation problem – Solution 1



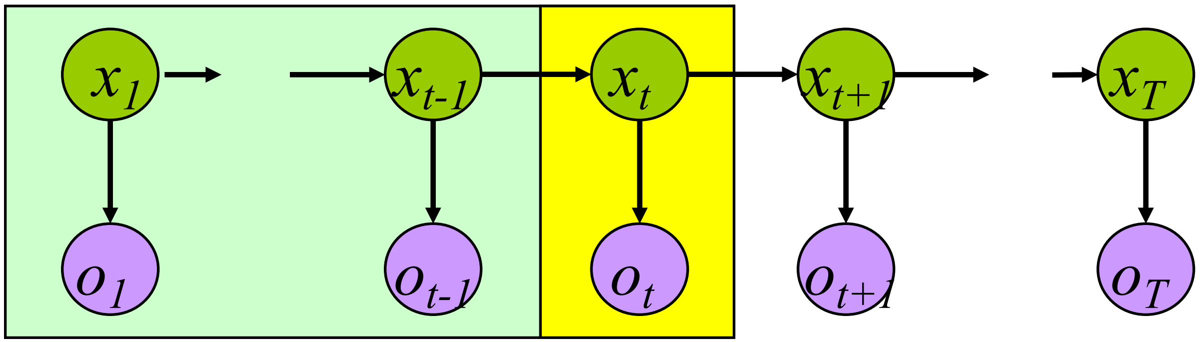
$$P(O \mid \mu) = ?$$

$$P(O \mid \mu) = \sum_X P(O \mid X, \mu) P(X \mid \mu)$$

$$P(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

- List all possible state sequences (length = T)
- Computational complexity： $2TN^T$ operations (totally N possible states)

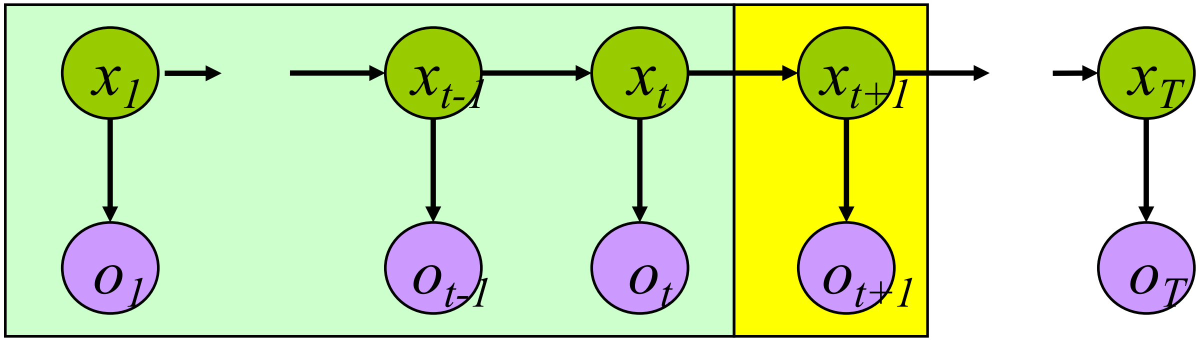
Estimation problem – Solution 2: Forward algorithm



- Special structure gives us an efficient solution using *dynamic programming* – forward algorithm.
- Define:

$$\boxed{\alpha_t(i) = P(o_1 \dots o_t, x_t = i \mid \mu)}$$

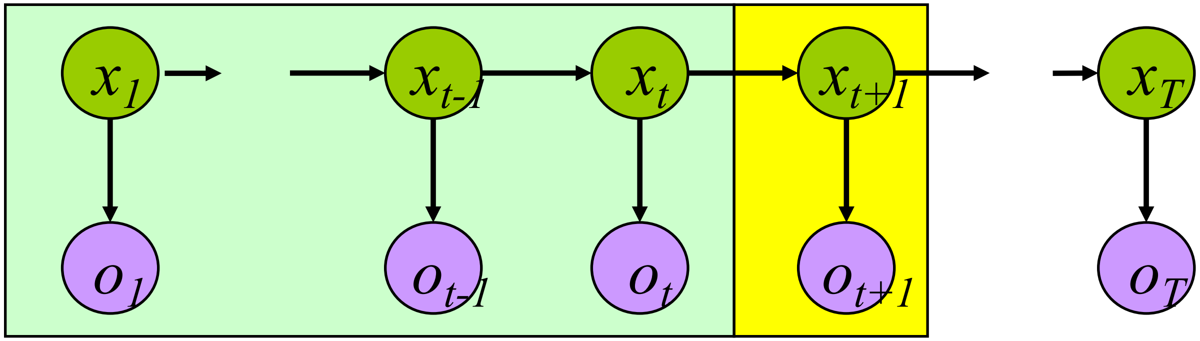
Forward algorithm



$$\alpha_t(i) = P(o_1 \ldots o_t, x_t = i \mid \mu)$$

$$\alpha_{t+1}(j) \;\; = P(o_1 \ldots o_{t+1}, x_{t+1} \;\; = j)$$

Forward algorithm

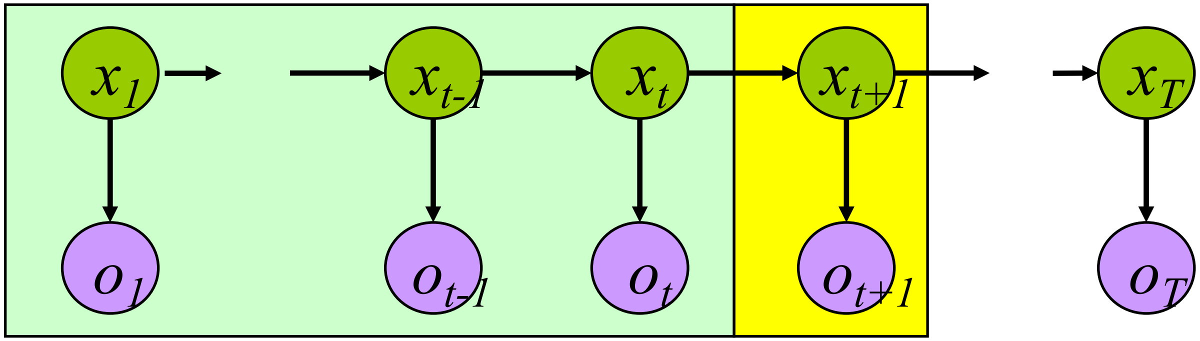


$$\boxed{\alpha_t(i) = P(o_1 \ldots o_t, x_t = i \mid \mu)}$$

$$\alpha_{t+1}(j) = P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

$$\boxed{= \sum_{i=1\ldots N} P(o_1 \ldots o_t, o_{t+1}, x_t = i, x_{t+1} = j)}$$

Forward algorithm



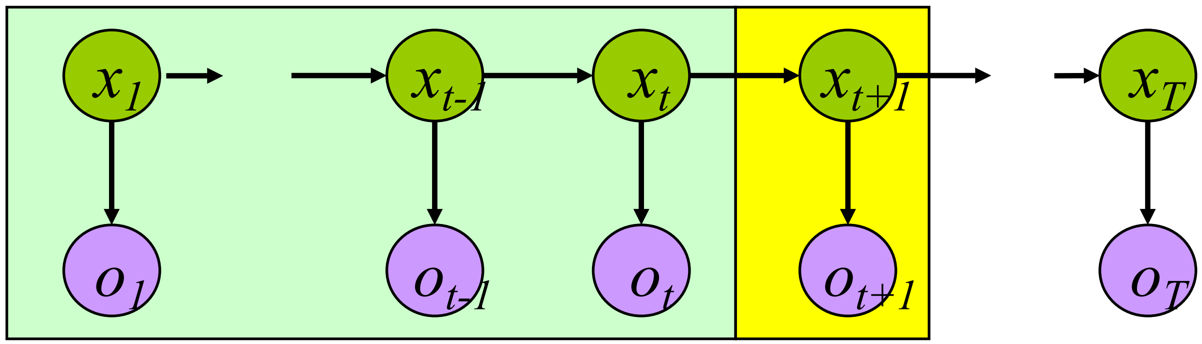
$$\boxed{\alpha_t(i) = P(o_1 \ldots o_t, x_t = i \mid \mu)}$$

$$\alpha_{t+1}(j) = P(o_1 \ldots o_{t+1}, x_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, o_{t+1}, x_t = i, x_{t+1} = j)$$

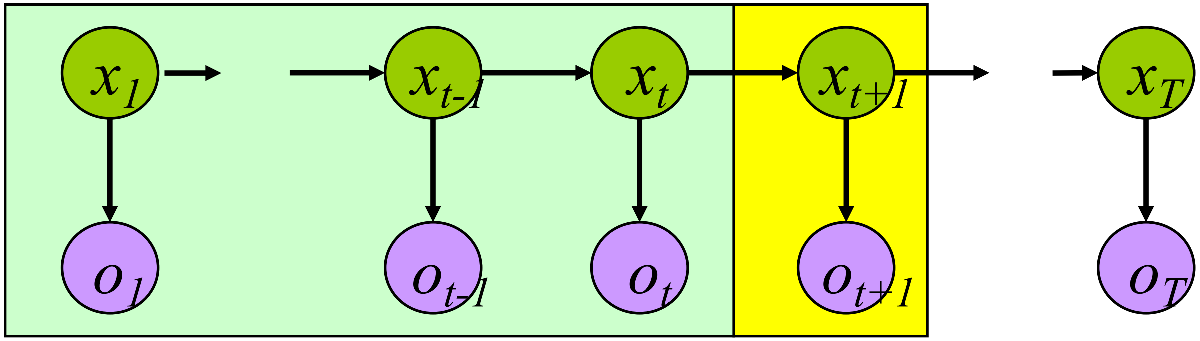
$$= \sum_{i=1 \ldots N} P(o_1 \ldots o_t, o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)$$

Forward algorithm



$$\alpha_{t+1}(j) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)$$

Forward algorithm

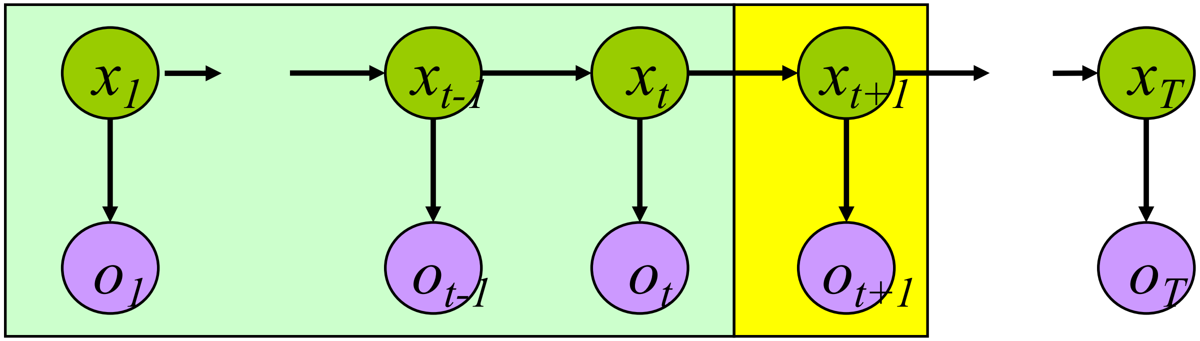


$$\alpha_{t+1}(j) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)$$

Independent assumption

$$\boxed{= \sum_{i=1\ldots N} P(o_1 \ldots o_t \mid x_t = i) P(o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)}$$

Forward algorithm

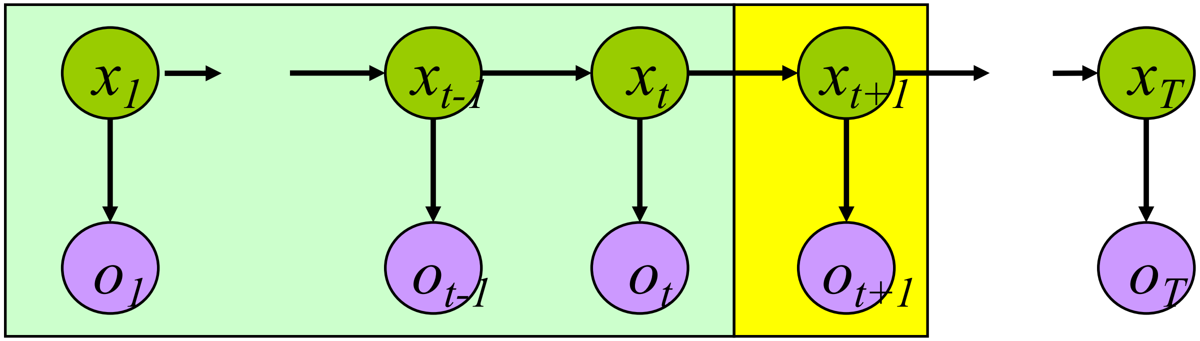


$$\alpha_{t+1}(j) = \sum_{i=1\ldots N} P(o_1 \ldots o_t, o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)$$

$$= \sum_{i=1\ldots N} P(o_1 \ldots o_t \mid x_t = i) P(o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i)$$

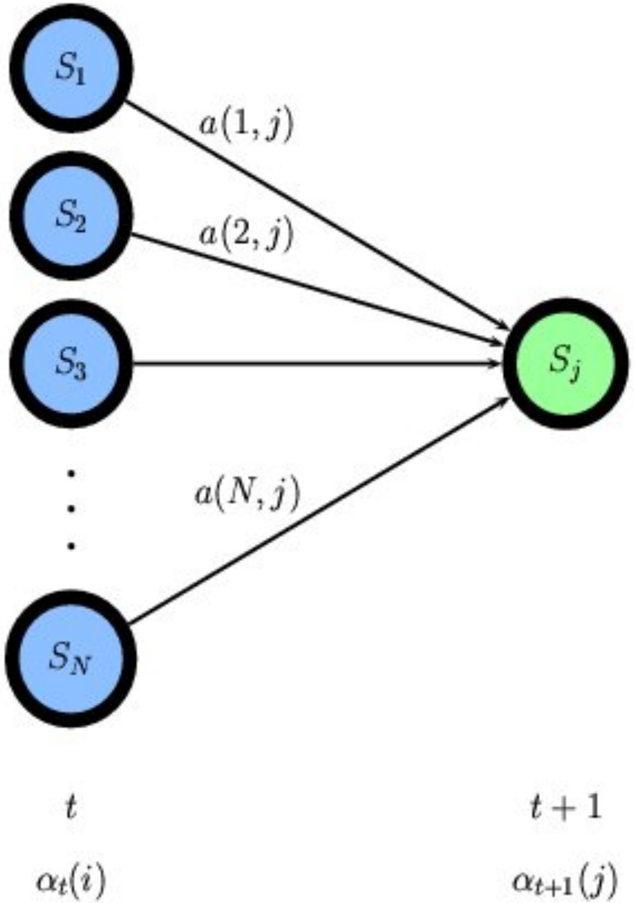
$$\boxed{= \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i) P(x_{t+1} = j \mid x_t = i) P(o_{t+1} \mid x_{t+1} = j)}$$

Forward algorithm



$$\begin{aligned}
\alpha_{t+1}(j) &= \sum_{i=1\ldots N} P(o_1 \ldots o_t, o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i) \\
&= \sum_{i=1\ldots N} P(o_1 \ldots o_t \mid x_t = i) P(o_{t+1}, x_{t+1} = j \mid x_t = i) P(x_t = i) \\
&= \sum_{i=1\ldots N} P(o_1 \ldots o_t, x_t = i) P(x_{t+1} = j \mid x_t = i) P(o_{t+1} \mid x_{t+1} = j) \\
&= \boxed{\sum_{i=1\ldots N} \alpha_t(i) a_{ij} b_{j o_{t+1}}}
\end{aligned}$$

Forward algorithm



$$\boxed{P(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)}$$

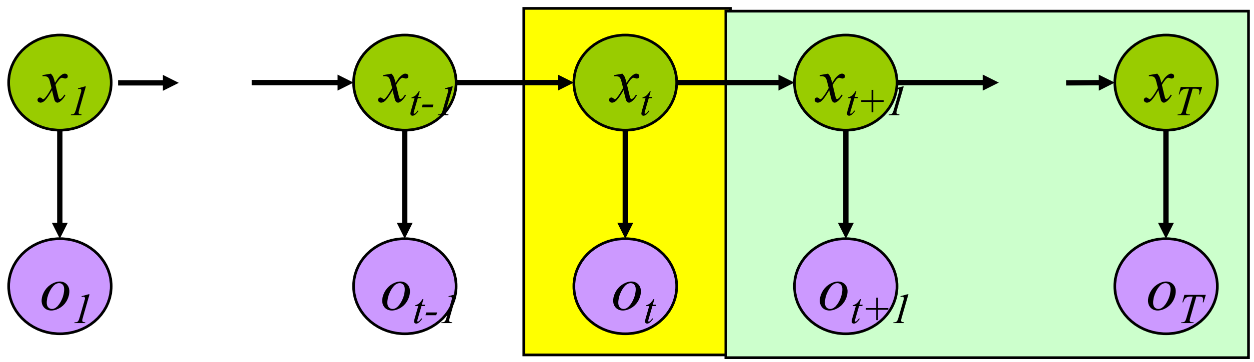
$$\alpha_{t+1}(j) = \sum_{i=1 \ldots N} \alpha_t(i) a_{ij} b_{jo_{t+1}}$$

- **N^2T operations**

(totally N possible states)

Estimation problem

– Solution 3: Backward algorithm

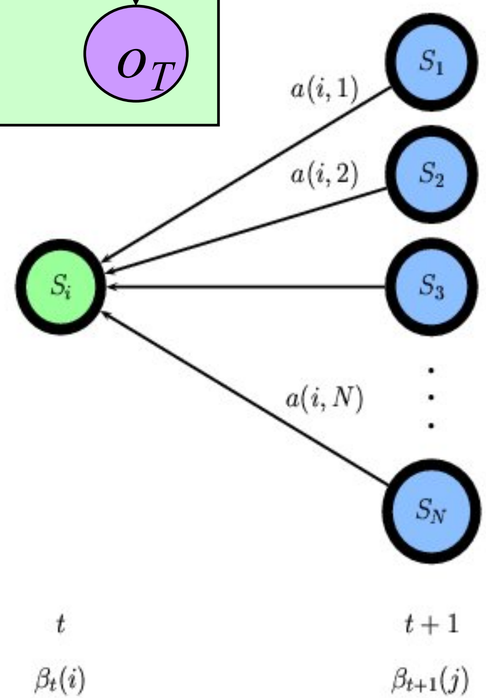


$$\alpha_t(i) = P(o_1 \ldots o_t, x_t = i \mid \mu)$$

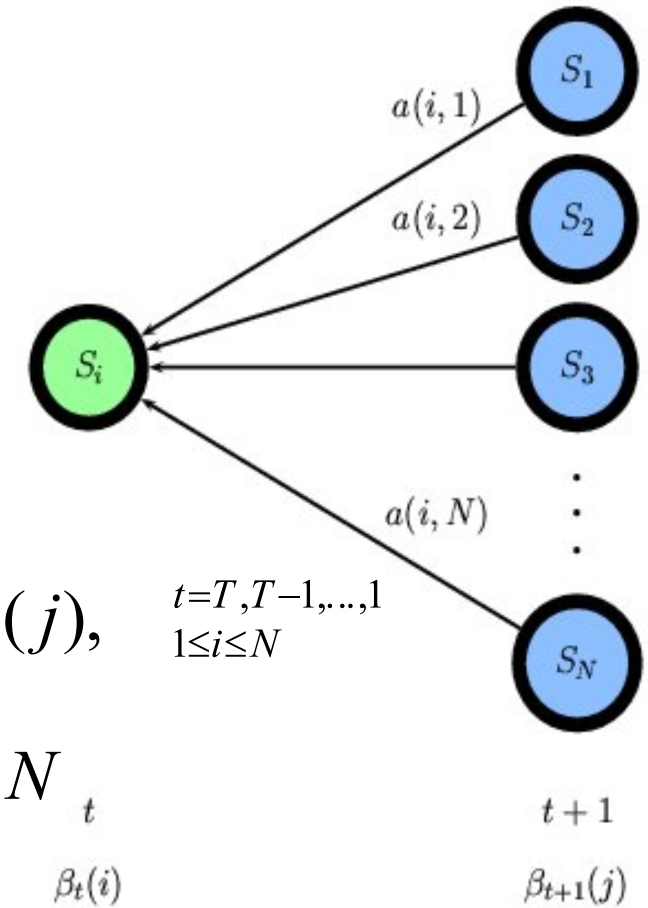
$$\beta_t(i) = P(o_t \ldots o_T \mid x_t = i, \mu)$$

$$\beta_t(i) = \sum_{j=1\ldots N} a_{ij} b_{io_t} \beta_{t+1}(j), \qquad \begin{matrix} t=T,T-1,\ldots,1 \\ 1 \le i \le N \end{matrix}$$

$$\beta_{T+1}(i) = 1, \quad 1 \le i \le N$$



Backward algorithm

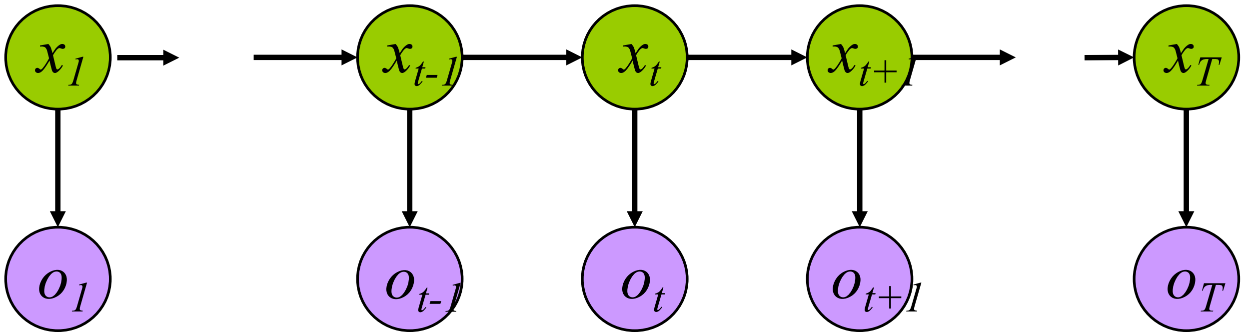


$$P(O \mid \mu) = \sum_{i=1}^{N} \pi_i \beta_1(i)$$

$$\beta_t(i) = \sum_{j=1\ldots N} a_{ij} b_{io_t} \beta_{t+1}(j), \quad \begin{array}{l} t=T,T-1,\ldots,1 \\ 1 \le i \le N \end{array}$$

$$\beta_{T+1}(i) = 1, \quad 1 \le i \le N$$

Estimation problem – Overview



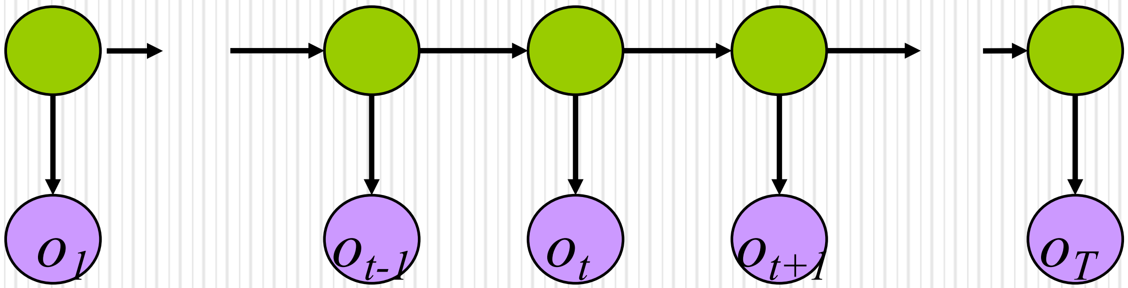
$$P(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$ **Enumeration method**

$$P(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$ **Forward algorithm**

$$P(O \mid \mu) = \sum_{i=1}^{N} \pi_i \beta_1(i)$$ **Backward algorithm**

Solution to Problem 2 – Encoding problem

Given an observation sequence, compute the most likely hidden state sequence

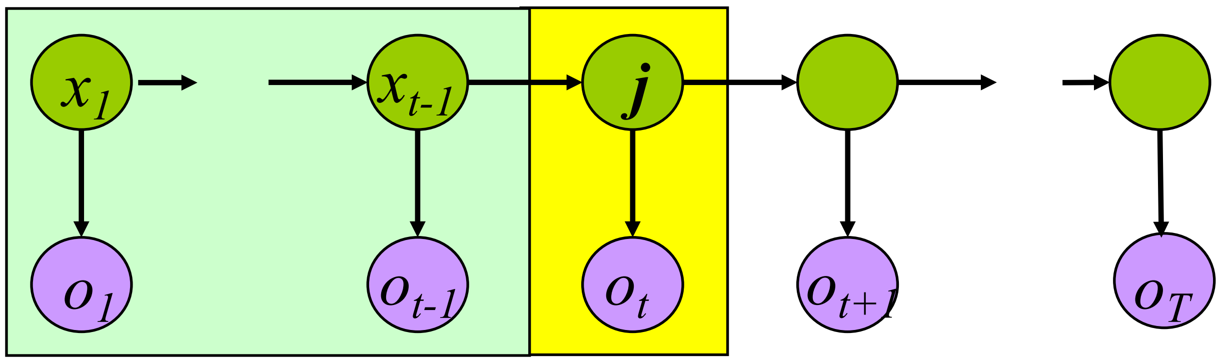


Problem 2 – Solution 2

- Given an observation sequence, compute the most likely hidden state sequence
- Find the state sequence that best explains the observations
- There may be many X' s that make P(X|O) maximal.
- We give an algorithm to find one of them.

$$\arg\max_X P(X \mid O) \longrightarrow \textbf{Viterbi algorithm}$$

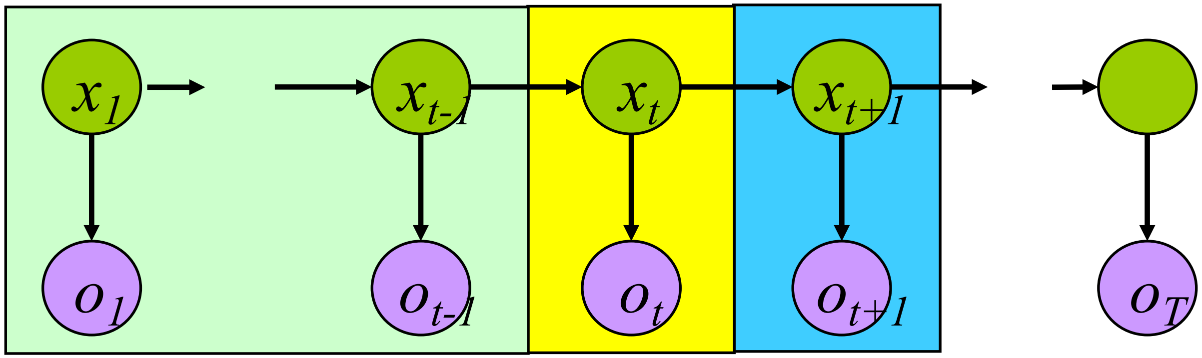
Viterbi Algorithm



$$\delta_t(j) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

The state sequence which maximizes the probability of seeing the observations to time t -1, landing in state j, and seeing the observation at time t

Viterbi Algorithm



$$\delta_t(j) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

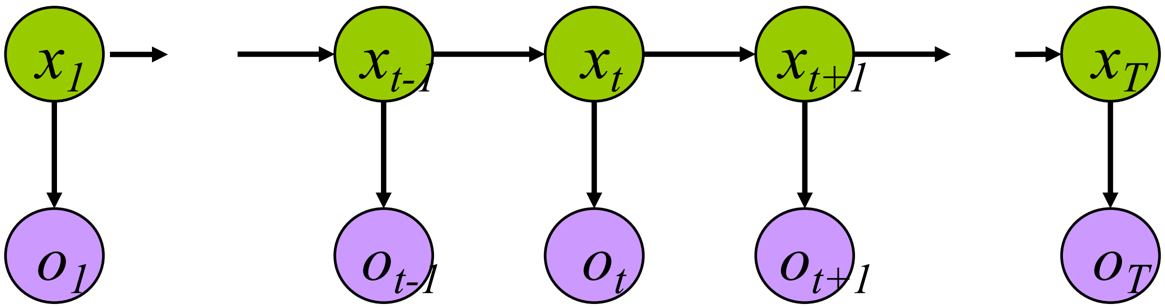
$$\delta_{t+1}(j) = \max_i \{\delta_t(i) a_{ij} b_{j o_{t+1}}\}$$

$$\psi_{t+1}(j) = \arg\max_i \{\delta_t(i) a_{ij} b_{j o_{t+1}}\}$$

Recursive

Computation

Viterbi Algorithm



$$\delta_t(j) = \max_{x_1 \ldots x_{t-1}} P(x_1 \ldots x_{t-1}, o_1 \ldots o_{t-1}, x_t = j, o_t)$$

$$\delta_{t+1}(j) = \max_i \{\delta_t(i) a_{ij} b_{j o_{t+1}}\} \qquad \psi_{t+1}(j) = \arg\max_i \{\delta_t(i) a_{ij} b_{j o_{t+1}}\}$$

$$\hat{X}_T = \arg\max_i \delta_T(i)$$

$$P(\hat{X}) = \delta_T(i)$$

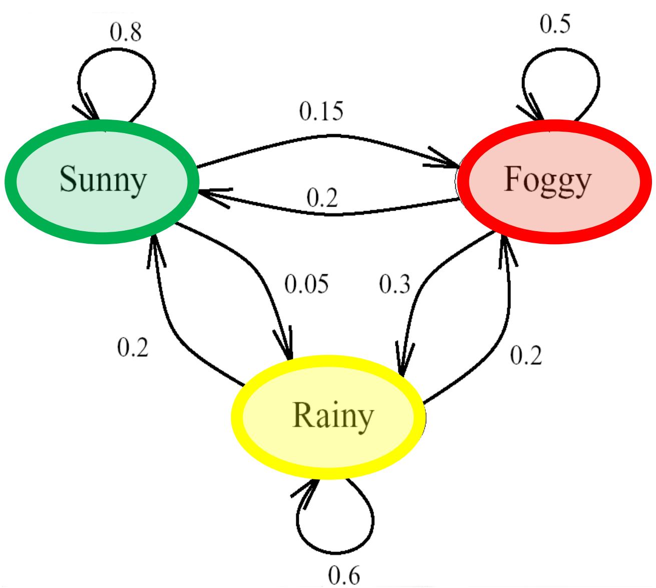
HMM applications

- Speech recognition
- (Chinese / Japanese) Input Method
- POS (Part of Speech) Tagging
- Gene analysis
- Any phenomena of linear sequence

Appendix

Appendix: one of the Answers to Question 3

$$
\begin{aligned}
p(x_2 = S_2 | y_2 = C_1, x_1 = S_1) &= \frac{p(x_2 = S_2, y_2 = C_1, x_1 = S_1)}{p(y_2 = C_1, x_1 = S_1)} \\
&= \frac{p(x_2 = S_2 | x_1 = S_1) p(y_2 = C_1 | x_2 = S_2) p(x_1 = S_1)}{\sum_{x_2} p(y_2 = C_1, x_2, x_1 = S_1)} \\
&= \frac{p(x_2 = S_2 | x_1 = S_1) p(y_2 = C_1 | x_2 = S_2) p(x_1 = S_1)}{\sum_{x_2} p(y_2 = C_1 | x_2) p(x_2 | x_1 = S_1) p(x_1 = S_1)} \\
&= \frac{p(x_2 = S_2 | x_1 = S_1) p(y_2 = C_1 | x_2 = S_2)}{\sum_{x_2} p(y_2 = C_1 | x_2) p(x_2 | x_1 = S_1)} \\
&= \frac{0.05 \cdot 0.8}{0.1 \cdot 0.8 + 0.8 \cdot 0.05 + 0.3 \cdot 0.15} = 0.243.
\end{aligned}
$$



	With umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

		Tomorrow's Weather		
		Sunny	Rainy	Foggy
Today's Weather	Sunny	0.8	0.05	0.15
	Rainy	0.2	0.6	0.2
	Foggy	0.2	0.3	0.5