1

Hidden Markov Model

Wei-Ta Chu

Markov Model

Chain rule

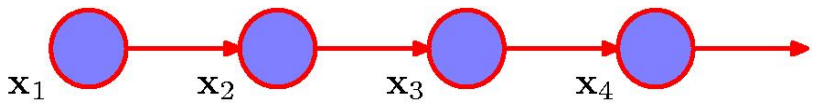
$$P(A, B, C) = P(A|B, C)P(B|C)P(C)$$

- Assume that each of the condition distributions is independent of all previous observations except the most recent, we obtain the *first-order Markov chain*.

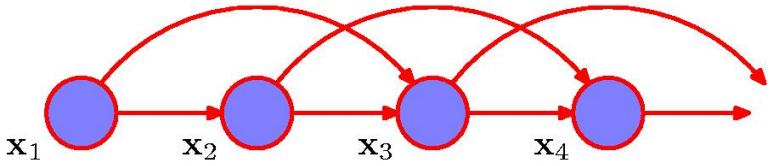
$$p(x_1, ..., x_N) = \prod_{n=1}^{N} p(x_n|x_1, ..., x_{n-1})$$

$$\Rightarrow \quad p(x_1, ..., x_N) = p(x_1) \prod_{n=2}^{N} p(x_n|x_{n-1})$$

$$p(x_n|x_1, ..., x_{n-1}) = p(x_n|x_{n-1})$$



First-order Markov chain



Second-order Markov chain

C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Example

- What's the probability that the weather for eight consecutive days is "sun-sun-sun-rain-rain-sun-cloudy-sun"?

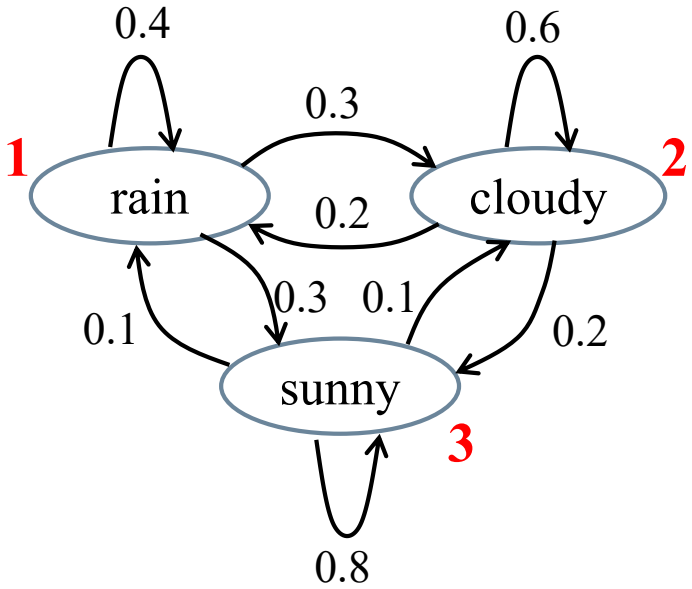
$$P(\boldsymbol{O}|Model) = P[3,3,3,1,1,3,2,3|Model]$$

$$= P[3]P[3|3]^2P[1|3]P[1|1]P[3|1]P[2|3]P[3|2]$$

$$= \pi_3(a_{33})^2a_{31}a_{11}a_{13}a_{32}a_{23}$$

$$= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2)$$

$$= 1.536 \times 10^{-4}$$



$$A = a_{ij} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, 1993
L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of IEEE, vol. 77, no. 2, pp. 257-286, 1989.

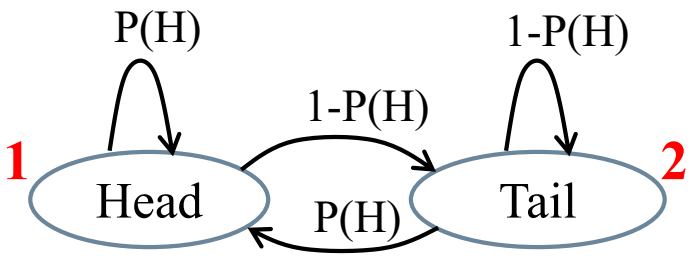
Coin-Toss Model

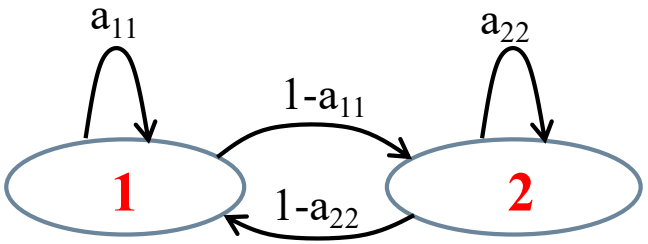
- You are in a room with a curtain through which you cannot see that is happening. On the other side of the curtain is another person who is performing a coin tossing experiment (using one or more coins). The person will not tell you which coin he selects at any time; he will only tell you the result of each coin flip.
- A typical observation sequence would be

$$\boldsymbol{O} = (\boldsymbol{o}_1 \boldsymbol{o}_2 \boldsymbol{o}_3 \cdots \boldsymbol{o}_T) = (H H T T T T T H T T T H \cdots H)$$

- The question is: how do we build an model to explain the observed sequence of head and tails?

Coin-Toss Model





P(H) = P_1 P(H) = P_2

P(T) =1- P_1 P(T) =1- P_2

1-coin model
(Observable Markov Model)

$$\boldsymbol{O} = (HHTTTTHTTH)$$

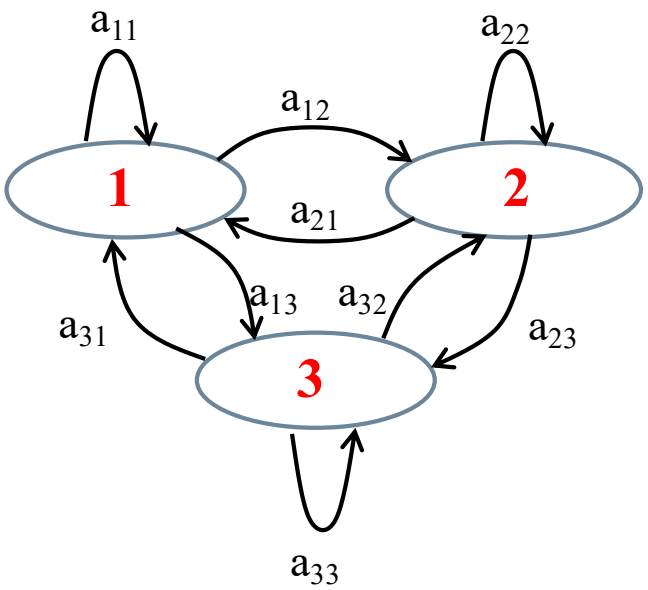
$$\boldsymbol{S} = (1122221221)$$

2-coins model
(Hidden Markov Model)

$$\boldsymbol{O} = (HHTTTTHTTH)$$

$$\boldsymbol{S} = (2112221221)$$

Coin-Toss Model



3-coins model
(Hidden Markov Model)

$$\boldsymbol{O} = (HHTTTTHTTH)$$

$$\boldsymbol{S} = (3123311231)$$

State 1	State 2	State 3
P(H) = P_1	P(H) = P_2	P(H) = P_3
P(T) =1- P_1	P(T) =1- P_2	P(T) =1- P_3

Elements of HMM

- $\boldsymbol{N}$: the number of states in the model
- $\boldsymbol{M}$: the number of distinct observation symbols per state
- The state-transition probability $\boldsymbol{A}=\{a_{ij}\}$

$$a_{ij} = P[q_{t+1} = j | q_t = i] \qquad 1 \le i, j \le N$$

- The observation symbol probability distribution $\boldsymbol{B}=\{b_j(k)\}$

$$b_j(k) = P[\boldsymbol{o}_t = \boldsymbol{v}_k | q_t = j] \qquad 1 \le k \le M$$

- The initial state distribution

$$\pi_i = P[q_1 = i] \qquad 1 \le i \le N$$

- To describe an HMM, we usually use the compact notation

$$\lambda = (A, B, \pi)$$

Three Basic Problems of HMM

- **Problem 1**: Probability Evaluation

Given the observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1 \boldsymbol{o}_2 \cdots \boldsymbol{o}_T)$, and a model $\lambda = (A, B, \pi)$, how do we efficiently computer $P(\boldsymbol{O}|\lambda)$, the probability of the observation sequence, given the model?

How do we compute the probability that the observed sequence was produced by the model?

Scoring how well a given model matches a given observation sequence.

Three Basic Problems of HMM

- **Problem 2**: Optimal State Sequence

  Given the observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1 \boldsymbol{o}_2 \cdots \boldsymbol{o}_T)$, and the model λ, how do we choose a corresponding state sequence $\boldsymbol{q} = (q_1 q_2 \cdots q_T)$ that is optimal in some sense (i.e., best explains the observations)

Attempt to uncover the hidden part of the model – that is, to find the "correct" state sequence.

For practical situations, we usually use an optimality criterion to solve this problem as best as possible.

Three Basic Problems of HMM

- **Problem 3**: Parameter Estimation

How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(\boldsymbol{O}|\lambda)$

Attempt to optimize the model parameters to best describe how a given observation sequence comes about.

The observation sequence used to adjust the model parameters is called a training sequence because it is used to "train" the HMM.

Solution to Problem 1

Given the observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1 \boldsymbol{o}_2 \cdots \boldsymbol{o}_T)$, and a model $\lambda = (A, B, \pi)$, how do we efficiently computer $P(\boldsymbol{O}|\lambda)$, the probability of the observation sequence, given the model?

- There are N^T possible state sequences
- Consider one fixed-state sequence $\boldsymbol{q} = (q_1 q_2 \cdots q_T)$
- The prob. of the observation sequence given the sequence

$$P(\boldsymbol{O}|\boldsymbol{q}, \lambda) = \prod_{t=1}^{T} P(\boldsymbol{o}_t|q_t, \lambda)$$

- Where we have assumed statistical independence of observations, thus we get

$$P(\boldsymbol{O}|\boldsymbol{q}, \lambda) = b_{q_1}(\boldsymbol{o}_1) \cdot b_{q_2}(\boldsymbol{o}_2) \cdots b_{q_T}(\boldsymbol{o}_T)$$

Solution to Problem 1

Solution to Problem 1

- The prob. of such state sequence can be written as

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

- The joint prob. of $\boldsymbol{O}$ and $\boldsymbol{q}$, i.e., the prob. That $\boldsymbol{O}$ and $\boldsymbol{q}$ occur simultaneously, is simply the product of the above terms

$$P(\boldsymbol{O}, \boldsymbol{q}|\lambda) = P(\boldsymbol{O}|\boldsymbol{q}, \lambda) P(\boldsymbol{q}|\lambda)$$

- The prob. of $\boldsymbol{O}$ is obtained by summing this joint prob. over all possible state sequences $\boldsymbol{q}$, giving

$$\begin{aligned} P(\boldsymbol{O}|\lambda) &= \sum_{all\boldsymbol{q}} P(\boldsymbol{O}|\boldsymbol{q}, \lambda) P(\boldsymbol{q}|\lambda) \\ &= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(\boldsymbol{o}_1) a_{q_1 q_2} b_{q_2}(\boldsymbol{o}_2) \cdots a_{q_{T-1} q_T} b_{q_T}(\boldsymbol{o}_T) \end{aligned}$$

Solution to Problem 1

- **The Forward Procedure**
- The prob. of the partial observation sequence $\boldsymbol{o}_1,\boldsymbol{o}_2, \ldots,\boldsymbol{o}_t$ (until time t) and state i at time t, given the model λ

$$\alpha_t(i) = P(\boldsymbol{o}_1\boldsymbol{o}_2 \cdots \boldsymbol{o}_t, q_t = i|\lambda)$$

- We solve for it inductively
- 1. Initialization $\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1) \qquad 1 \le i \le N$
- 2. Induction

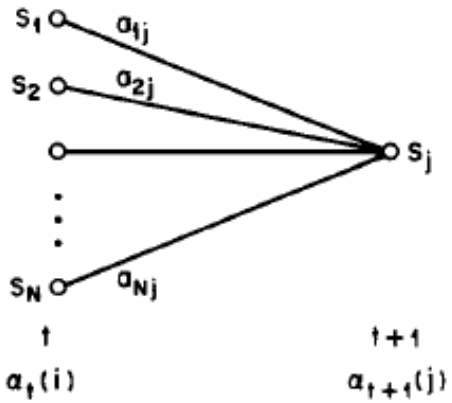
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(\boldsymbol{o}_{t+1}) \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix}$$

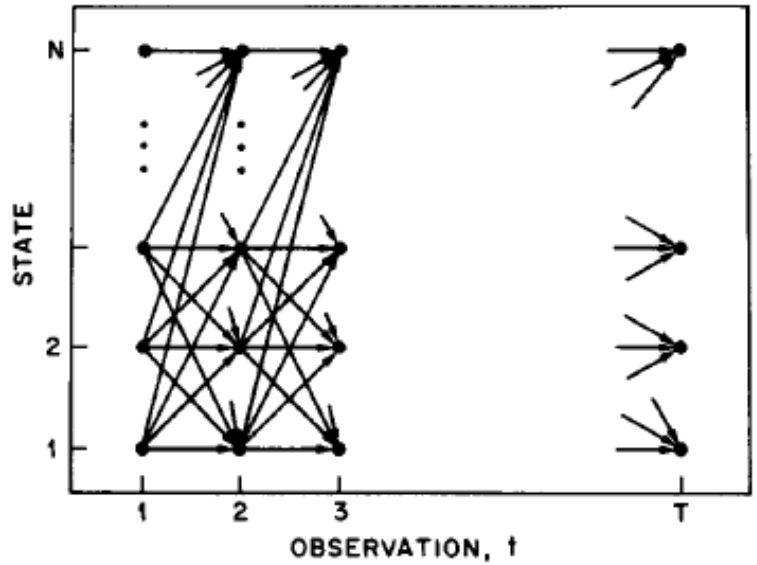
- 3. Termination

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Solution to Problem 1

- **The Forward Procedure**





- Require on the order of N^2T calculations, rather than $2TN^T$ as required by the direction calculation.

- **The Backward Procedure**
- The prob. of partial observation sequence from t+1 to the end, given state i at time t and the model λ

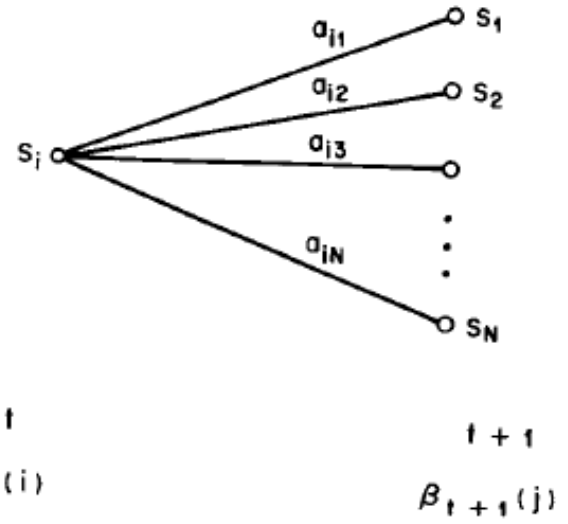
$$\beta_t(i) = P(\boldsymbol{o}_{t+1}\boldsymbol{o}_{t+2}\cdots\boldsymbol{o}_T|q_t = i, \lambda)$$

- 1. Initialization $\beta_T(i) = 1 \qquad 1 \le i \le N$
- 2. Induction

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

$$t = T-1, T-2, ..., 1$$

$$1 \le i \le N$$



Solution to Problem 2

Given the observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1\boldsymbol{o}_2 \cdots \boldsymbol{o}_T)$, and the model λ, how do we choose a corresponding state sequence $\boldsymbol{q} = (q_1 q_2 \cdots q_T)$ that is optimal in some sense (i.e., best explains the observations)

- We define the quantity

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i, \boldsymbol{o}_{t+1}\boldsymbol{o}_{t+2} \cdots \boldsymbol{o}_t | \lambda]$$

- Which is the best score (highest probability) along a single path, at time *t*, which accounts for the first *t* observations and ends in state *i*. By induction we have

$$\delta_{t+1}(i) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(\boldsymbol{o}_{t+1})$$

Solution to Problem 2

Solution to Problem 2

- **The Viterbi Algorithm**
- 1. Initialization $\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1) \qquad \psi_1(i) = 0$
- 2. Recursion $\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i)a_{ij}]b_j(\boldsymbol{o}_t) \qquad 1 \le j \le N$
- $\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i)a_{ij}] \qquad 2 \le t \le T$
- 3. Termination $P^* = \max_{1 \le i \le N}[\delta_T(i)]$

  $q_T^* = \arg\max_{1 \le i \le N}[\delta_T(i)]$
- 4. Path (state sequence) backtracking

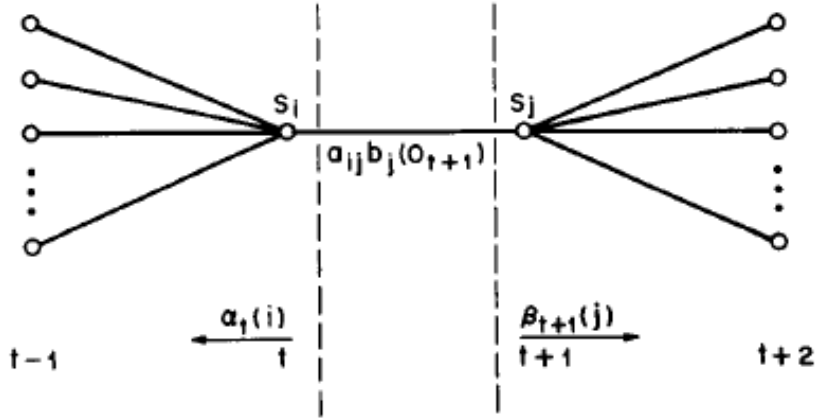
  $q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, ..., 1$
- The major difference between Viterbi and the forward procedure is the maximization over previous states.

Solution to Problem 3

How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(\boldsymbol{O}|\lambda)$

- Choose $\lambda = (A, B, \pi)$such that its likelihood, $P(\boldsymbol{O}|\lambda)$, is locally maximized using an iterative procedure such as the Baum-Welch algorithm (also known as EM algorithm or forward-backward algorithm)
- Define the prob. of being in state *i* at time *t*, and state *j* at time *t*+1, given the model and the observation sequence.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\boldsymbol{O}, \lambda)$$



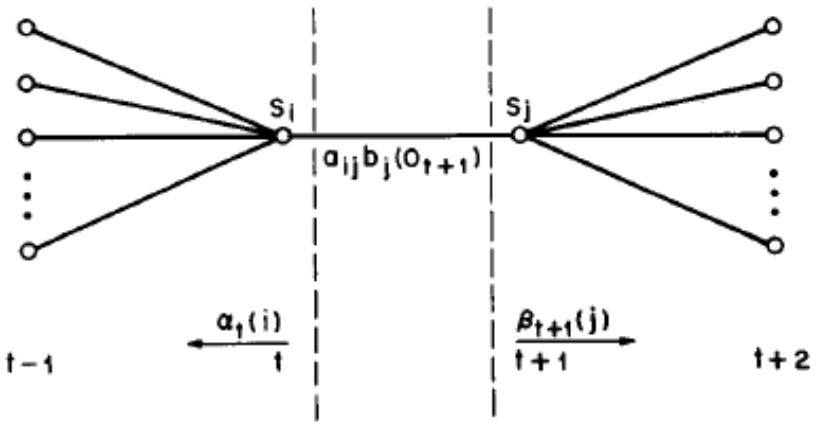
Solution to Problem 3

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | \boldsymbol{O}, \lambda)$$

$$\xi_t(i,j) = \frac{P(q_t{=}i, q_{t+1}{=}j, \boldsymbol{O}|\lambda)}{P(\boldsymbol{O}|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{P(\boldsymbol{O}|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}$$



$$\gamma_t(i) = P(q_t = i | \boldsymbol{O}, \lambda)$$

The prob. of being in state *i* at time *t*, given the observation sequence ***O*** and the model

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

$$\gamma_t(i) = P(q_t = i | \boldsymbol{O}, \lambda) = \frac{P(\boldsymbol{O}, q_t{=}i|\lambda)}{P(\boldsymbol{O}|\lambda)}$$

$$= \frac{P(\boldsymbol{O}, q_t{=}i|\lambda)}{\sum_{i=1}^{N} P(\boldsymbol{O}, q_t{=}i|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

Solution to Problem 3

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

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

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } i \text{ at time } (t=1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

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

$$= \frac{\sum_{\substack{t=1 \\ s.t. \boldsymbol{o}_t = \boldsymbol{v}_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

→ Usually modeled by GMM

Types of HMMs

- Ergodic
- Left-right
- Parallel path left-right

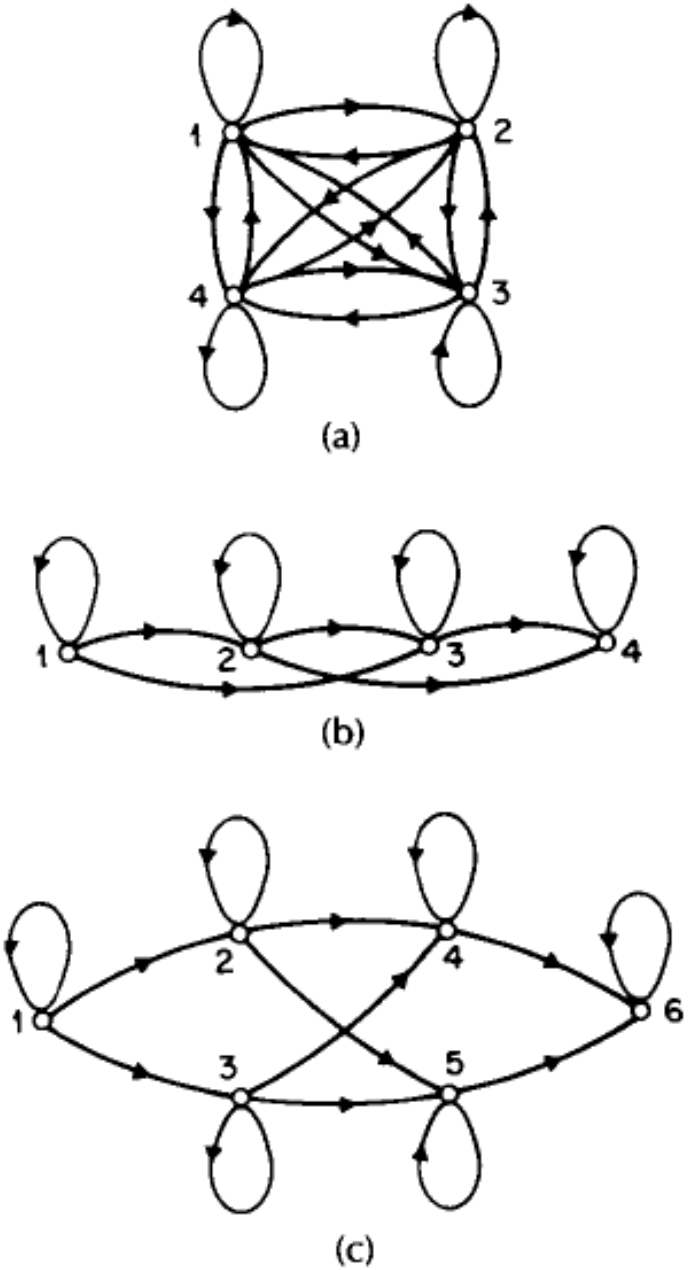


Fig. 7. Illustration of 3 distinct types of HMMs. (a) A 4-state ergodic model. (b) A 4-state left-right model. (c) A 6-state parallel path left-right model.

Case Study

- Features
  - Field descriptor
  - Edge descriptor
  - Grass and sand
  - Player height
- For each highlight model, compute the optimal state sequence by the Viterbi algorithm.
- Check the prob. derived from the state sequence.

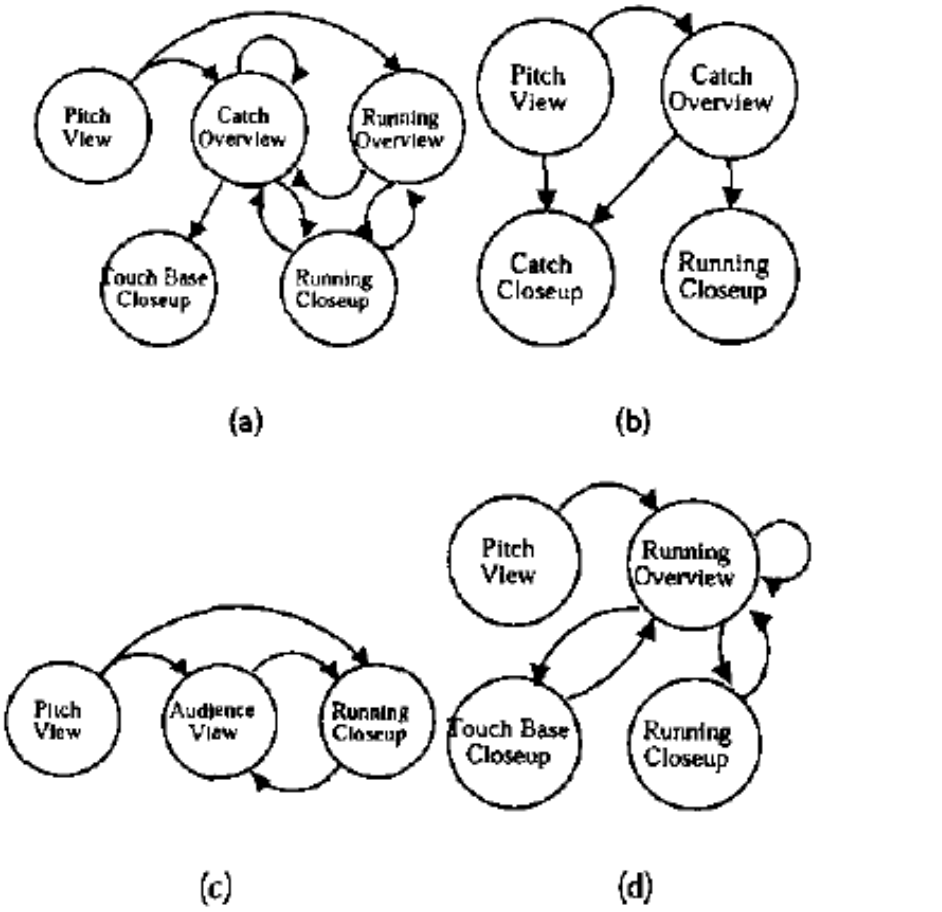


Fig. 2. (a) HMM model for nice hits (b) HMM model for nice catches (c) HMM model for home runs (d) HMM model for plays within the diamond

Peng, et al. "Extract highlights from baseball game video with hidden Markov models," In Proc. of ICIP, vol. 1, pp. 609-612, 2002.

Related Resources

- Hidden Markov Model (HMM) Toolbox for Matlab
  - http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
- The General Hidden Markov Model library (GHMM)
  - http://ghmm.sourceforge.net/
- HTK Speech Recognition Toolkit
  - http://htk.eng.cam.ac.uk