《机器学习》课件

3.2隐性马氏模型（HMM）及应用

HMM的由来

- 1870年，俄国有机化学家Vladimir V. Markovnikov第一次提出马尔科夫模型
  - 马尔可夫模型
  - 马尔可夫链
  - 隐马尔可夫模型

马尔可夫性

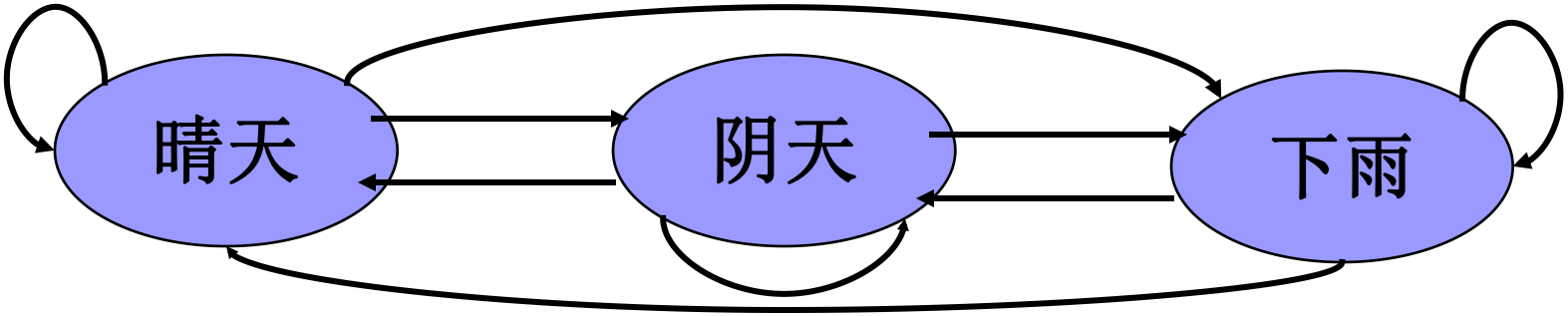
- 如果一个过程的“将来”仅依赖“现在”而不依赖“过去”，则此过程具有**马尔可夫性**,或称此过程为**马尔可夫过程**

- $X(t+1) = f(X(t))$

马尔科夫链

- **时间**和**状态**都离散的马尔科夫过程称为马尔科夫链
- 记作$\{X_n = X(n), n = 0,1,2,\ldots\}$
  - 在时间集$T_1 = \{0,1,2,\ldots\}$上对离散状态的过程相继观察的结果
- 链的状态空间记做$I = \{a_1, a_2,\ldots\}, a_i \in R.$
- 条件概率$P_{ij}(m, m+n) = P\{X_{m+n} = a_j | X_m = a_i\}$ 为马氏链在时刻m处于状态a_i条件下，在时刻m+n转移到状态a_j的**转移概率**。

转移概率矩阵



	晴天	阴天	下雨
晴天	0.50	0.25	0.25
阴天	0.375	0.25	0.375
下雨	0.25	0.125	0.625

马尔可夫性质

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

- {'Dry','Dry','Rain',Rain'}.
  P({'Dry','Dry','Rain',Rain'}) = ?

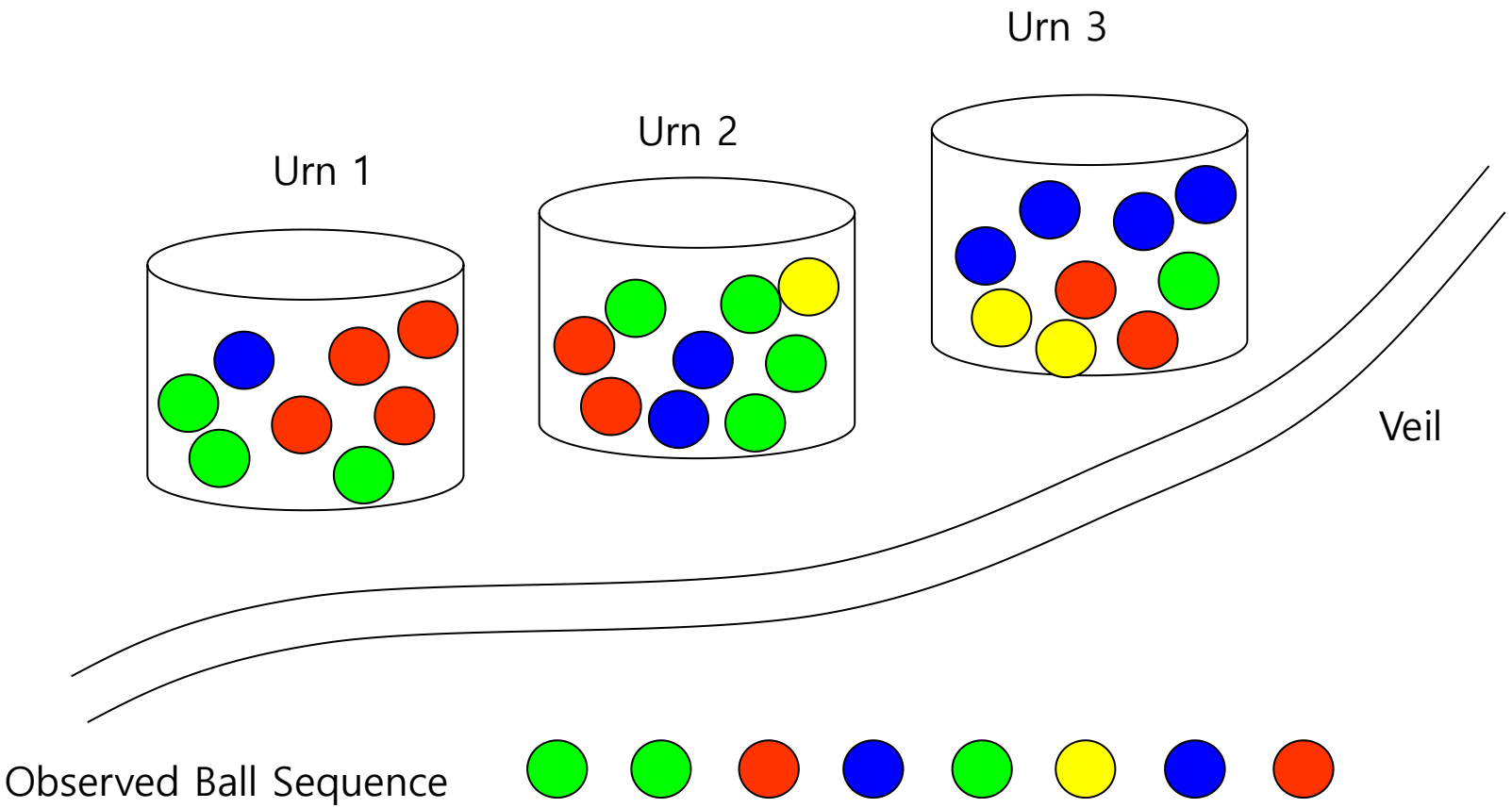
转移概率矩阵(续)

- 由于链在时刻m从任何一个状态a_i出发，到另一时刻m+n，必然转移到a_1，a_2…，诸状态中的某一个，所以有

$$\sum_{j=1}^{\infty} P_{ij}(m, m+n) = 1, i = 1, 2, \ldots$$

- 当P_{ij}(m,m+n)与m无关时，称马尔科夫链为**齐次马尔科夫链**，通常说的马尔科夫链都是指齐次马尔科夫链。

HMM实例



HMM实例——描述

- 设有N个缸，每个缸中装有很多彩球，球的颜色由一组概率分布描述。实验进行方式如下
  - 根据初始概率分布，随机选择N个缸中的一个开始实验
  - 根据缸中球颜色的概率分布，随机选择一个球，记球的颜色为O_1，并把球放回缸中
  - 根据描述缸的转移的概率分布，随机选择下一口缸，重复以上步骤。
- 最后得到一个描述球的颜色的序列$O_1,O_2,\ldots,$称为观察值序列O。

HMM实例——约束

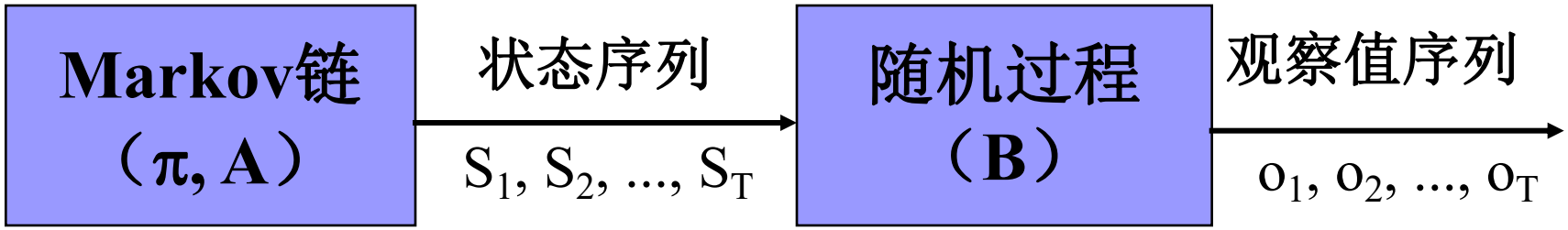
在上述实验中，有几个要点需要注意：

- 不能被直接观察杯子间的转移
- 从缸中所选取的球的颜色和杯子并不是 一一对应的
- 每次选取哪个杯子由一组转移概率决定

HMM概念

- HMM的状态是不确定或不可见的，只有通过观测序列的随机过程才能表现出来
- 观察到的事件与状态并不是一一对应，而是通过一组概率分布相联系
- HMM是一个双重随机过程，两个组成部分：
  - 马尔可夫链：描述状态的转移，用转移概率描述。
  - 一般随机过程：描述状态与观察序列间的关系，用观察值概率描述。

HMM组成



HMM的组成示意图

马氏过程与马氏链

- 马氏过程：具有无后效性的随机过程。即t_m时刻所处状态的概率只和t_{m-1}时刻的状态有关，而与t_{m-1}时刻之前的状态无关。比如布朗运动，柏松过程。
- 马氏链：时间离散，状态离散的马氏过程。

马氏链的参数

- 转移概率：$a_{kl}=P(S_i=l|\pi S_{i-1}=k)$

$$0 \le a_{kl} \le 1$$

$$\sum_{l} a_{kl} = 1$$

- 初始概率

HMM（Hidden Markov Models）

- 一个双重随机过程，两个组成部分：
  - 马氏链：描述状态的转移。
    用转移概率 a_{kl} 描述
  - 一般随机过程：描述状态与观察序列间的关系
    用输出概率 $e_k(b)$ 描述

HMM的基本算法

- Viterbi算法
- 前向 - 后向算法
- Baum-Welch算法

Viterbi算法

- 采用动态规划算法。复杂度O(K^2L)
  K和L分别为状态个数和序列长度
- 初始化（i = 0）：$v_0(0)=1, v_k(0)=0$　k>0

递推（i = 1…L）：$v_l(i)=e_l(x_i)\max_k\left(v_k(i-1)a_{kl}\right)$

$$ptr_i(l)=\arg\max_k\left(v_k(i-1)a_{kl}\right)$$

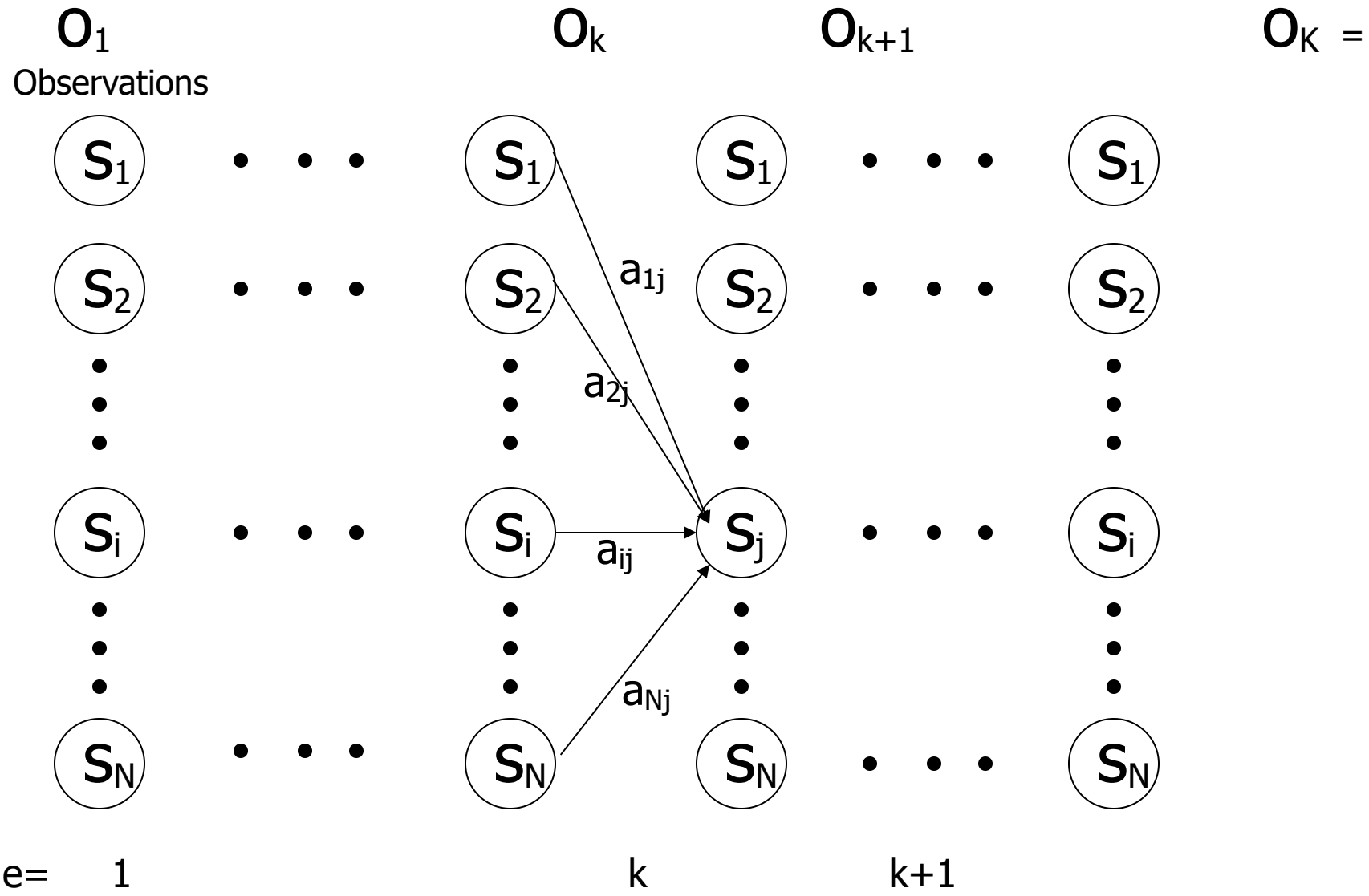
终止：

$$P(x,\pi^*)=\max_k\left(v_k(L)a_{k0}\right)$$

$$\pi_L^*=\arg\max_k\left(v_k(L)a_{k0}\right)$$

回溯（i = L…1）：$\pi_{i-1}^*=ptr_i(\pi_i^*)$

Trellis representation of an HMM



前向 - 后向算法

前向算法：动态规划，复杂度同Viterbi

定义前向变量 $f_k(i) = P(x_1 \ldots x_i, \pi_i = k)$

初始化（i = 0）：$f_0(0) = 1, f_k(0) = 0$ k>0

递推（i = 1…L）：$f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$

终止：$P(x) = \sum_k f_k(L) a_{k0}$

后向算法：动态规划，复杂度同Viterbi

定义后向变量 $b_k(i) = P(x_{i+1}\ldots x_L \mid \pi_i = k)$

初始化（i = L）：

$b_k(L) = a_{k0}$ 所有 k

递推（i = L - 1…1）：

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$

终止：

$$P(x) = \sum_l a_{0l} e_l(x_1) b_l(1)$$

Learning problem

- If training data has information about sequence of hidden states (as in word recognition example), then use maximum likelihood estimation of parameters:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Number of transitions from state } S_j \text{ to state } S_i}{\text{Number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Number of times observation } V_m \text{ occurs in state } S_i}{\text{Number of times in state } S_i}$$

Baum-Welch algorithm

General idea:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Expected number of transitions from state } S_j \text{ to state } S_i}{\text{Expected number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Expected number of times observation } V_m \text{ occurs in state } S_i}{\text{Expected number of times in state } S_i}$$

$$\pi_i = P(s_i) = \text{Expected frequency in state } S_i \text{ at time } k=1.$$

Baum - Welch算法

- 重估公式:

$$A_{kl} = \sum_j \frac{1}{p(x^j)} \sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)$$

$$E_k(b) = \sum_j \frac{1}{p(x^j)} \sum_{\{i|x_i^j=b\}} f_k^j(i) b_k^j(i)$$

HMM应用

- 主要应用是解码（decoding）。

  在生物序列分析中，从序列中的每个值（观察值） 去推测它可能属于那个状态。
- 两种解码方法：

  （1）Viterbi算法解码

  （2）前向 - 后向算法 + 贝叶斯后验概率

Viterbi解码

- 由Viterbi算法所得的是一条最佳路径。根据该路径可直接得出对应于每一观察值的状态序列

前向 - 后向算法 + 贝叶斯后验概率

- 利用贝叶斯后验概率计算序列中的值属于某一状态的概率即：

$$P(\pi_i = k \mid x) = \frac{P(x, \pi_i)}{P(x)}$$

而

$$P(x, \pi_i) = P(x_1 .. x_i, \pi_i = k) P(x_{i+1} .. x_L \mid \pi_i = k)$$
$$= f_k(i) b_k(i)$$

实际建模过程

- 根据实际问题确定状态个数及观察序列。
- 用若干已知序列，采用B－W算法估计参数（转移概率 a_{ij} 和输出概率 $b_k(b)$ 的值。
- 输入未知序列用Viterbi算法或贝叶斯概率解码。

例

假设你有三个好朋友A、B、C，因为学习繁忙，每周只能抽出一天时间陪他们中的一个吃饭或看电影。娱乐活动结束后，你通常会发一条朋友圈表达喜悦，为了保护朋友的隐私，你不会在朋友圈里说明和谁出去玩，只会说今天玩了什么（吃饭/看电影）。

例

这三个朋友并非完全相同，你对他们的好感度也有所区别。于是你在心中确定了陪伴这三位朋友的概率。

朋友	A	B	C
概率	0.2	0.4	0.4

例

如果这一周你陪伴了某一个朋友，你很可能意犹未尽，下一周还想和他一起玩，所以本周和谁玩还影响了下一周的选择。

本周/下周	A	B	C
A	0.5	0.2	0.3
B	0.3	0.5	0.2
C	0.2	0.3	0.5

例

这三个朋友爱好有所区别，你在选择这周做什么的时候通常会顾及朋友的想法。因此你心里给出了陪不同朋友时会做什么的概率。

朋友	吃饭	看电影
A	0.5	0.5
B	0.4	0.6
C	0.7	0.3

建立模型

朋友	A	B	C
概率	0.2	0.4	0.4

在这个例子中，第一张表是没有什何干扰情
况
初
下

本周/下周	A	B	C
A	0.5	0.2	0.3
B	0.3	0.5	0.2
C	0.2	0.3	0.5

模型的
的影响
态转移
矩阵。第三张表是你在某种状态下做某种事的概率，是模型的观测概率矩阵。

朋友	吃饭	看电影
A	0.5	0.5
B	0.4	0.6
C	0.7	0.3

建立模型

这个例子就可以用隐马尔可夫模型进行表达。

在外界看来（只能看朋友圈），只能看到你这一周做了什么（观测结果），而不清楚你陪了哪位朋友（实际状态）。你的状态是隐藏在后面的，这就是隐马尔可夫模型中“隐”的含义。

预测问题（Viterbi算法）

你每周发的朋友圈引起了你室友的兴趣。你的室友认识你的三个朋友，也知道你对他们的看法（即知道上面的三张表），但是你并不打算将你陪谁出去玩告诉室友，这引起了他们的好奇心。他们想根据朋友圈的信息和三张表推断出你每周都去陪了谁。

预测问题（Viterbi算法）

假设你的室友从朋友圈得知，你前三周分别去吃饭、看电影、吃饭。用标号0表示陪A朋友，标号1表示陪B朋友，标号2表示陪C朋友。标号0表示观测结果为“吃饭”，1表示观测结果为“看电影”。s表示初始状态向量。

第0周（t=0）：

$$v_0(0) = s_0 e_0(0) = 0.2 \times 0.5 = 0.1$$

$$v_1(0) = s_1 e_1(0) = 0.4 \times 0.4 = 0.16$$

$$v_2(0) = s_2 e_2(0) = 0.4 \times 0.7 = 0.28$$

预测问题（Viterbi算法）

开始递推，第一周时（t=0）

$$v_0(1) = \max_{0 \le j \le 2}\left(v_j(0)a_{j0}\right) e_0(1) = \max\{0.025, 0.024, 0.028\} = 0.028$$

$$v_1(1) = \max_{0 \le j \le 2}\left(v_j(0)a_{j1}\right) e_1(1) = \max\{0.012, 0.048, 0.0504\} = 0.0504$$

$$v_2(1) = \max_{0 \le j \le 2}\left(v_i(0)a_{j2}\right) e_2(1) = \max\{0.009, 0.0096, 0.042\} = 0.042$$

预测问题（Viterbi算法）

第二周时（t=2）

$$v_0(2) = \max_{0\le j\le 2}\left(v_j(0)a_{j0}\right)e_0(1) = \max\{0.007, 0.00756, 0.0042\} = 0.00756$$

$$v_1(2) = \max_{0\le j\le 2}\left(v_j(0)a_{j1}\right)e_1(1)$$

$$= \max\{0.00224, 0.01008, 0.00504\} = 0.01008$$

$$v_2(2) = \max_{0\le j\le 2}\left(v_j(0)a_{j2}\right)e_2(1) = \max\{0.00588, 0.007056, 0.0147\} = 0.0147$$

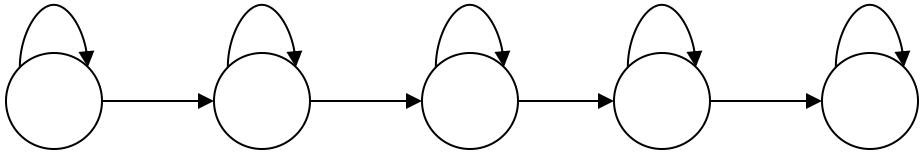
预测问题（Viterbi算法）

- 反向推导最优隐状态序列：
- $\pi_2 = 2$
- $\pi_1 = \underset{0 \le j \le 2}{\operatorname{argmax}}(v_j(1)\, a_{j2}) = 2$
- $\pi_0 = \underset{0 \le j \le 2}{\operatorname{argmax}}(v_j(0)\, a_{j2}) = 2$

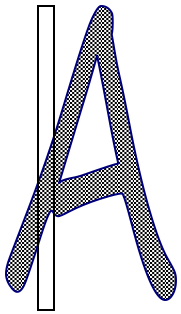
因此，预测结果为这三周你都在陪朋友C。

Character recognition with HMM example.

- The structure of hidden states is chosen.



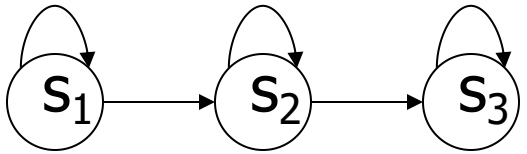
- Observations are feature vectors extracted from vertical slice



- Probabilistic mapping from hidden state to feature vectors:
    1. use mixture of Gaussian models
    2. Quantize feature vector space.

Exercise: character recognition with HMM(1)

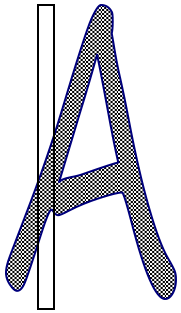
- The structure of hidden states:



- Observation = number of islands in the vertical slice.
- HMM for character 'A' :

Transition probabilities: $\{a_{ij}\} = \begin{pmatrix} .8 & .2 & 0 \\ 0 & .8 & .2 \\ 0 & 0 & 1 \end{pmatrix}$

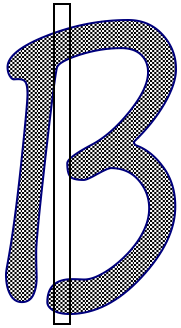
Observation probabilities: $\{b_{jk}\} = \begin{pmatrix} .9 & .1 & 0 \\ .1 & .8 & .1 \\ .9 & .1 & 0 \end{pmatrix}$



- HMM for character 'B' :

Transition probabilities: $\{a_{ij}\} = \begin{pmatrix} .8 & .2 & 0 \\ 0 & .8 & .2 \\ 0 & 0 & 1 \end{pmatrix}$

Observation probabilities: $\{b_{jk}\} = \begin{pmatrix} .9 & .1 & 0 \\ 0 & .2 & .8 \\ .6 & .4 & 0 \end{pmatrix}$



Exercise: character recognition with HMM(2)

- Suppose that after character image segmentation the following sequence of island numbers in 4 slices was observed:

  { 1, 3, 2, 1}

- What HMM is more likely to generate this observation sequence , HMM for ‘A’ or HMM for ‘B’ ?

Exercise: character recognition with HMM(3)

Consider likelihood of generating given observation for each possible sequence of hidden states:

- HMM for character 'A':

Hidden state sequence	Transition probabilities	Observation probabilities
$S_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$	$.8 * .2 * .2 = 0$	$* .9 * 0 * .8 * .9$
$S_1 \rightarrow S_2 \rightarrow S_2 \rightarrow S_3$	$.2 * .8 * .2 = 0.0020736$	$* .9 * .1 * .8 * .9$
$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3$	$.2 * .2 * 1 = 0.000324$	$* .9 * .1 * .1 * .9 =$ Total = 0.0023976

- HMM for character 'B':

Hidden state sequence	Transition probabilities	Observation probabilities
$S_1 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$	$.8 * .2 * .2 = 0$	$* .9 * 0 * .2 * .6$
$S_1 \rightarrow S_2 \rightarrow S_2 \rightarrow S_3$	$.2 * .8 * .2 = 0.0027648$	$* .9 * .8 * .2 * .6$
$S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_3$	$.2 * .2 * 1 = 0.006912$	$* .9 * .8 * .4 * .6 =$ Total = 0.0096768

Evaluation Problem.

- **Evaluation problem.** Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1 o_2 \ldots o_K$, calculate the probability that model M has generated sequence O.
- Trying to find probability of observations $O=o_1 o_2 \ldots o_K$ by means of considering all hidden state sequences (as was done in example) is impractical:

  N^K hidden state sequences - exponential complexity.

- Use **Forward-Backward HMM algorithms** for efficient calculations.

- Define the forward variable $\alpha_k(i)$ as the joint probability of the partial observation sequence $o_1 o_2 \ldots o_k$ and that the hidden state at time k is s_i : $\alpha_k(i)= P(o_1 o_2 \ldots o_k, q_k= s_i)$