

第2章：隐马氏模型(HMM) 及其应用

- Markov Model
- Page rank算法和Markov Model
- Hidden Markov Model (HMM)
- HMM的理论基础
- HMM的应用

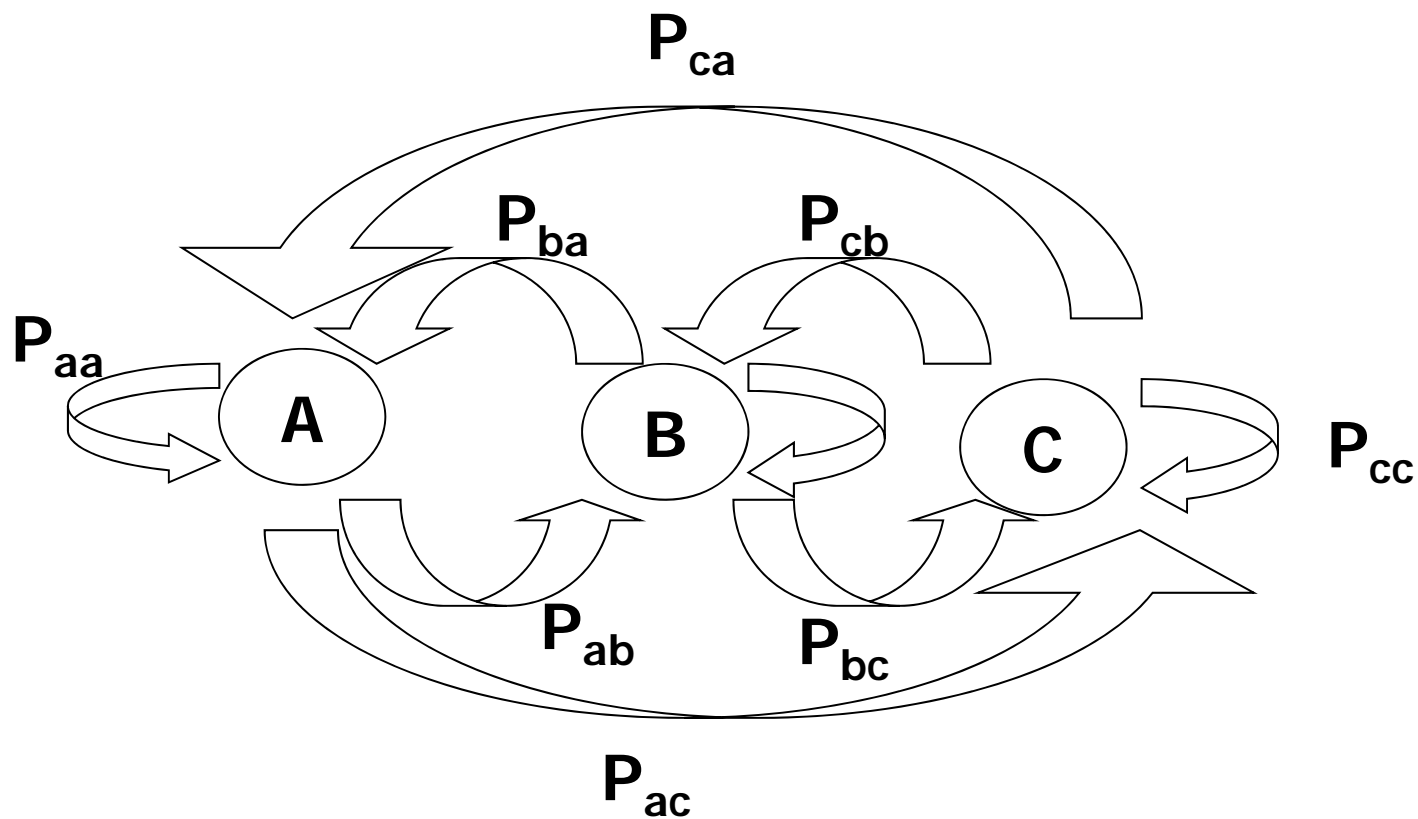
Part I

Markov Model

食堂就餐人数问题 (I)

- 某大学有三个食堂A、B、C。调查显示：在食堂A就餐的人中 p_{aa} 部分仍然回到食堂A，有 p_{ab} 部分选择食堂B， p_{ac} 部分选择食堂C；在食堂B就餐的人中 p_{bb} 部分仍然回到食堂B，有 p_{ba} 部分选择食堂A， p_{bc} 部分选择食堂C；在食堂C就餐的人中 p_{cc} 部分仍然回到食堂C，有 p_{ca} 部分选择食堂A， p_{cb} 部分选择食堂B；
- 请估计在食堂A、B、C的就餐人数。

食堂就餐人数问题 (II)



食堂就餐人数问题 (III)

- 令 A_n 为第 n 天在食堂A就餐的人数比例
- 令 B_n 为第 n 天在食堂B就餐的人数比例
- 令 C_n 为第 n 天在食堂C就餐的人数比例

$$A_{n+1} = p_{aa}A_n + p_{ba}B_n + P_{ca}C_n$$

$$B_{n+1} = p_{ab}A_n + p_{bb}B_n + P_{cb}C_n$$

$$C_{n+1} = p_{ac}A_n + p_{bc}B_n + P_{cc}C_n$$

不动点问题 (I)

$$\begin{aligned} & (A_{n+1}, B_{n+1}, C_{n+1}) \\ &= (A_n, B_n, C_n) \begin{pmatrix} p_{aa} & p_{ab} & p_{ac} \\ p_{ba} & p_{bb} & p_{bc} \\ p_{ca} & p_{cb} & p_{cc} \end{pmatrix} \end{aligned}$$

问题：极限是否存在？若存在，

$$(x, y, z) = (x, y, z) \begin{pmatrix} p_{aa} & p_{ab} & p_{ac} \\ p_{ba} & p_{bb} & p_{bc} \\ p_{ca} & p_{cb} & p_{cc} \end{pmatrix}$$

不动点问题(II)

- 若初值为 $\pi_0=(A_0, B_0, C_0)$ ，并令 P 为上式中的矩阵，则

$$(A_n, B_n, C_n) = \pi_0 P^n$$

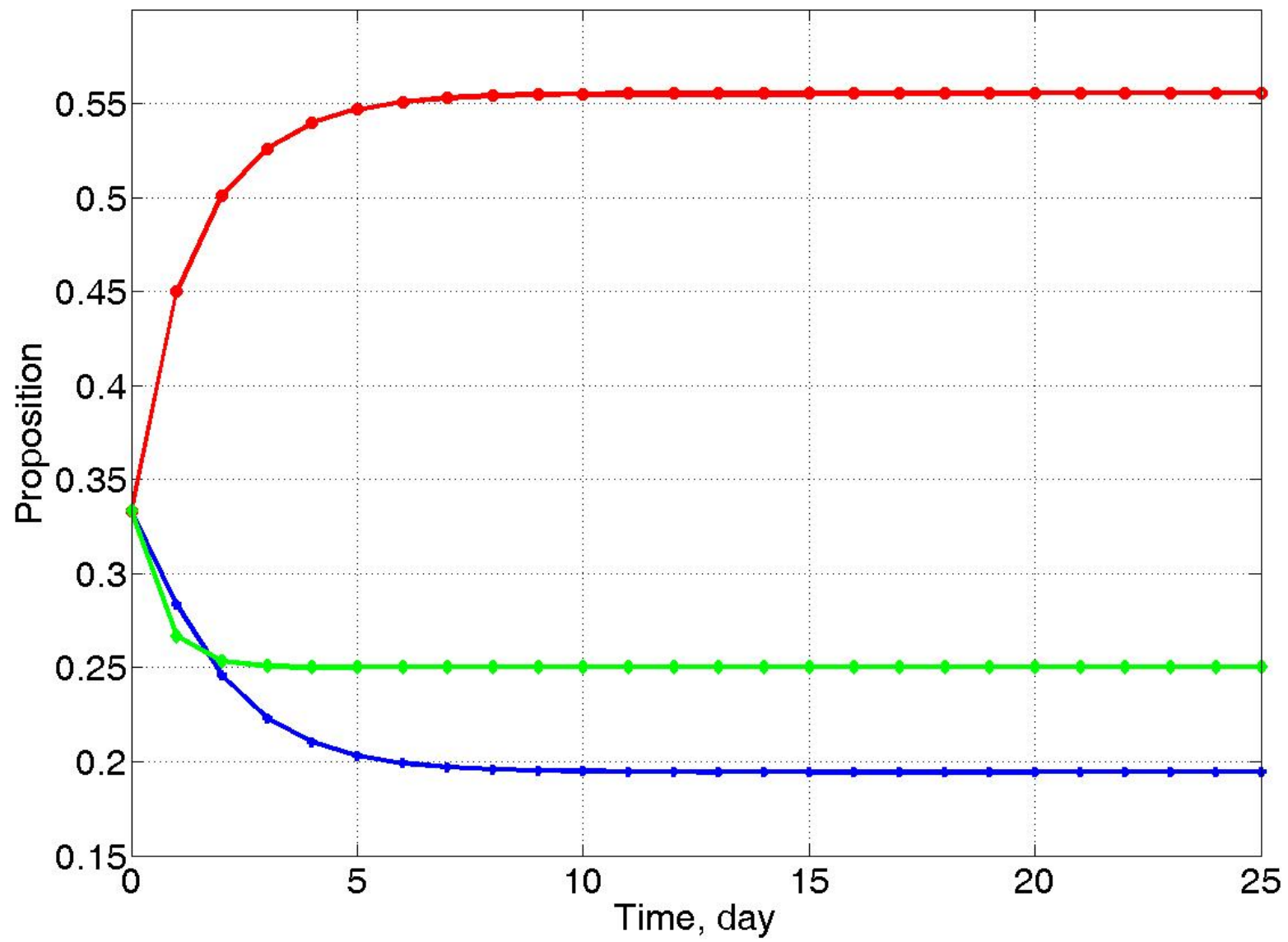
- 若 $\pi=(1/3, 1/3, 1/3)$ ， P 由下面的矩阵给出，我们可以具体计算 (A_n, B_n, C_n)

$$\begin{pmatrix} 0.75 & 0.05 & 0.20 \\ 0.20 & 0.60 & 0.20 \\ 0.40 & 0.20 & 0.40 \end{pmatrix}$$

不动点问题(III)

n	A_n	B_n	C_n	n	A_n	B_n	C_n
1	0.4500	0.2833	0.2667	11	0.5553	0.1947	0.2500
2	0.5008	0.2458	0.2533	12	0.5554	0.1946	0.2500
3	0.5261	0.2232	0.2507	13	0.5555	0.1945	0.2500
4	0.5395	0.2104	0.2501	14	0.5555	0.1945	0.2500
5	0.5468	0.2032	0.2500	15	0.5555	0.1945	0.2500
6	0.5507	0.1993	0.2500	16	0.5555	0.1945	0.2500
7	0.5529	0.1971	0.2500	17	0.5555	0.1945	0.2500
8	0.5541	0.1959	0.2500	18	0.5555	0.1945	0.2500
9	0.5548	0.1952	0.2500	19	0.5555	0.1945	0.2500
10	0.5551	0.1949	0.2500	20	0.5555	0.1945	0.2500

Dynamics of the numbers



换起点 $\pi=(0.8,0.1,0.1)$

n	A _n	B _n	C _n	n	A _n	B _n	C _n
1	0.66	0.12	0.22	11	0.5558	0.1942	0.25
2	0.607	0.149	0.244	12	0.5557	0.1943	0.25
3	0.5827	0.1686	0.2488	13	0.5556	0.1944	0.25
4	0.5702	0.1800	0.2498	14	0.5556	0.1944	0.25
5	0.5636	0.1865	0.2500	15	0.5556	0.1944	0.25
6	0.5600	0.1901	0.2500	16	0.5556	0.1944	0.25
7	0.5580	0.1920	0.2500	17	0.5556	0.1944	0.25
8	0.5569	0.1931	0.2500	18	0.5556	0.1944	0.25
9	0.5563	0.1937	0.2500	19	0.5556	0.1944	0.25
10	0.5560	0.1940	0.25	20	0.5556	0.1944	0.25

不动点还和前面的一样！

问题的特征

- 每一步活动只与当前处在什么“状态”有关，与过去的“状态”没有关系。
- 矩阵 P 特殊性：每行和为1，表示下一个时刻的状态必须在A、B、C中之一。
- 马尔可夫链模型，简称马氏链。

离散时间随机过程

- 对于离散的时间 $t=0, 1, 2, 3, \dots$ 的每一个 t 对应一个随机变量 $\xi_t(\omega)$ ，我们把 $\xi=\{\xi_0, \xi_1, \dots, \xi_n, \dots\}$ 这样一个随机变量的序列叫做离散时间的随机过程。
- 所有 $\xi_t(\omega) (t=0, 1, 2, 3, \dots)$ 具有公共的取值集合，我们把此集合叫做状态空间，记为 S 。

离散时间随机过程

- 对于一个固定的 ω , $\xi(\omega)=\{\xi_0(\omega), \xi_1(\omega), \cdots, \xi_n(\omega), \cdots\}$ 就是一个状态的序列, 称为该随机过程的一条轨道, 我们把 $\xi_t(\omega)$ 的取值叫做该条轨道在时间 t 的状态。
- $(\xi_{n_1}(\omega), \xi_{n_2}(\omega), \cdots, \xi_{n_m}(\omega))$ 的联合分布称为 ξ 的一个有限维分布, 我们用 ξ 的全部有限维分布刻画它的统计特性.

马氏(Markov)链

- 随机过程 $\{\xi_n(\omega), n > 0\}$ 称为有限状态马氏链, 若 ξ_n 只有有限个取值且满足

$$\begin{aligned} &P(\xi_{n+k} = j \mid \xi_n = i, \dots, \xi_0 = i_0) \\ &= P(\xi_{n+k} = j \mid \xi_n = i) \end{aligned}$$

- 记之为 $p_{ij}(n, n+k)$
- 矩阵 $P(n, n+k) = (p_{ij}(n, n+k))$ 称为从n出发的k步转移概率矩阵

时齐马氏链

- 如果马氏链的转移矩阵与出发时刻无关, 即 $P(n, n+k) = P(0, k)$, 则称此马氏链是时齐的。
- 这时将 $p_{ij}(n, n+k)$ 简单地记为 $p_{ij}(k)$.
- 通常不特别说明, 马氏链就指时齐马氏链。
- 前面的食堂问题就是一个1步时齐马氏链。

高阶马氏过程

- 若一个随机过程满足：

$$\begin{aligned} & \forall n; j, i_n, i_{n-1}, i_{n-2}, \cdots, i_0, \\ & P(x_{n+1} = j \mid x_n = i_n, \cdots, x_0 = i_0) \\ & = P(x_{n+1} = j \mid x_n = i_n, \cdots, x_{n-k+1} = i_{n-k+1}) \end{aligned}$$

也就是说随机过程下一时间的发展只和包括当前时间在内的最近的k个时间的状态有关 而和这k个时间之前的历史没有关系, (其中k=0, 1, 2, ...), 我们把这样的随机过程叫做k阶马氏链。

零(1)阶马氏过程

- 显然 零阶马氏链就是说下一时间的发展和当前状态及已有历史都独立，也就是相互独立的随机序列 (过程)。
- 1-阶马氏链就是前面的马氏链.

关于名称的一点说明

- 参考书中，看到马氏链(过程)的时候要根据上下文进行判断。有的时候是指普遍的马氏链(包括高阶、一阶、零阶)，有时候特指一阶马氏链。
- 在大多数情况下，如不特别说明，通常是特指一阶时齐的马氏链。
- 如果将一个 k -阶马氏链的相邻 k 个时间的状态合为一个新的状态: $y_n = (x_n, x_{n-1}, \dots, x_{n-k+1})$, 则 $\{y_n\}$ 是一个 1-阶马氏链。

转移概率矩阵性质

$$P(n, n) = I$$

$$P(n, m)1^T = 1^T$$

$$P(n, l) = P(n, m)P(m, l)$$

- 其中第三个方程称之为C-K方程

时齐马氏链性质 (I)

- 时齐马氏链由转移概率矩阵和初分布完全确定，设转移概率矩阵为 $P=(p_{ij})$ ，初始分布： $\mu_i^{(0)} = P(\xi_0 = i)$ 则

$$P = P(n, n+1)$$

$$P(n, n+m) = P^m$$

$$\begin{aligned} P(\xi_0 = i_0, \xi_1 = i_1, \dots, \xi_n = i_n) \\ = \mu_{i_0}^{(0)} p_{i_0, i_1} \cdots p_{i_{n-1}, i_n} \end{aligned}$$

时齐马氏链性质 (II)

- 若记

$$\mu_i^{(n)} = P(\xi_n = i), \mu^{(n)} = (\mu_i^{(n)}, i \in S),$$

即所谓绝对概率，则：

$$\mu^{(n+m)} = \mu^{(m)} P^n$$

马氏链的不变分布

- 状态空间 S 上的一个概率分布称为转移概率矩阵 P 的不变概率分布(简称不变分布)，如果

$$\pi P = \pi$$

- 一般来说，不变分布未必存在。若不变分布存在且唯一(记为 π)，则它是以下代数方程组的唯一非负解。

$$\pi = \pi P$$

$$\pi 1^T = 1^T$$

概念：可达、互通、不可约

- 可达：状态*i*称为可达状态*j*, 如果存在一个指标序列 $i_0 = i, i_1, \dots, i_n = j$, 使得

$$p_{i_k, i_{k+1}} > 0 \quad k = 0, 1, \dots, n-1.$$

- 用转移概率矩阵来刻画 *i*可达 *j*:

$$\exists n > 0, (p^n)_{ij} > 0.$$

- 互通：状态*i*可达状态*j*, 而且状态*j*可达状态*i*.
- 不可约：如果所有状态之间是互通的.

概念：常返

- 常返性：马氏链 $\{\xi_n(\omega) : n \geq 0\}$, 状态 y 称为常返的，如果概率为1地发生如下事件：从 y 出发的状态，有限时间内离开状态 y ，此后又必到达 y , \dots , 如此无限重复。

概念： 周期

- 从某个状态*i*出发, 经过*n*步回到状态*i*的情形。
为此定义集合

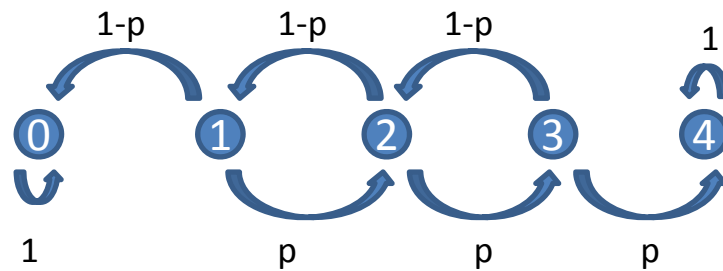
$$\{n : (p^n)_{ii} > 0\}$$

- 这个集合的最大公约数称之为状态*i*的周期*T*。
- 若*T*>1称状态*i*是周期的； 若*T*=1,称状态*i*为非周期的。

Example

- 状态1,2,3的周期为2

	0	1	2	3	4
0	1	0	0	0	0
1	$1-p$	0	p	0	0
2	0	$1-p$	0	p	0
3	0	0	$1-p$	0	p
4	0	0	0	0	1



马氏链的遍历极限(I)

若马氏链 $\{\xi_n(\omega) : n \geq 0\}$ 的状态空间 S 为有限集 (不妨设 $S=\{1,2,\cdots, N\}$), 且 ξ (转移矩阵为 P)是一个互通常返马氏链, 则它存在唯一的不变概率分 $\pi = (\pi_1, \pi_2, \cdots, \pi_N)$, 并使得

$$\lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n P^k}{n} = \pi 1^T$$

马氏链的遍历极限(II)

若马氏链 $\{\xi_n(\omega) : n \geq 0\}$ 的状态空间 S 为有限集(不妨设 $S = \{1, 2, \dots, N\}$), 且转移矩阵矩阵的每个元素为正, 则它存在唯一不变概率分布 $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_n)$, 满足如下(指数)遍历性

$$\sum_k \pi_k p_{kj} = \pi_j, \pi_j \geq \delta$$
$$|p_{ij}(n) - \pi_j| \leq (1 - N\delta)^n$$

马氏链的遍历极限(III)

- 令 $T_i(\omega)$ 是 $\{\xi_1(\omega), \dots, \xi_n(\omega)\}$ 中首次出现状态 i 的时间。那么 $\mu_i = E(T_i(\omega) | \xi_0(\omega) = i)$ 就是一个平均返回(状态 i)时间。有结论如下

对于互通常返马氏链

$$\mu_i = \frac{1}{\pi_i}$$

其中不变分布为 $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_n)$,

Part II

Page Rank算法和Markov Model

Page Rank

- Basic idea of Page Rank.
- Calculation of Page Rank.
- Page Rank and Markov chain

Basic Idea of Page Rank

- Suppose that:
 - Page d_1 contains hyperlinks to 10 pages;
 - Page d_2 just has a single link to page d_3
- The contribution of d_2 to $pr(d_3)$ should be greater than the contribution of d_1
- The simplest solution is to allocate a weight of $w_{de} = 1/L(d)$ to the hyperlink from document d to document e , where $L(d)$ is the number of hyperlinks from d .

Random Surfer Model

- w_{de} can be thought of as the probability of following the link to page e if the user is on page d .
- The case where $w_{de} = 1/L(d)$ corresponds to the random surfer model: on any page the random surfer is equally likely to choose any of the available links.

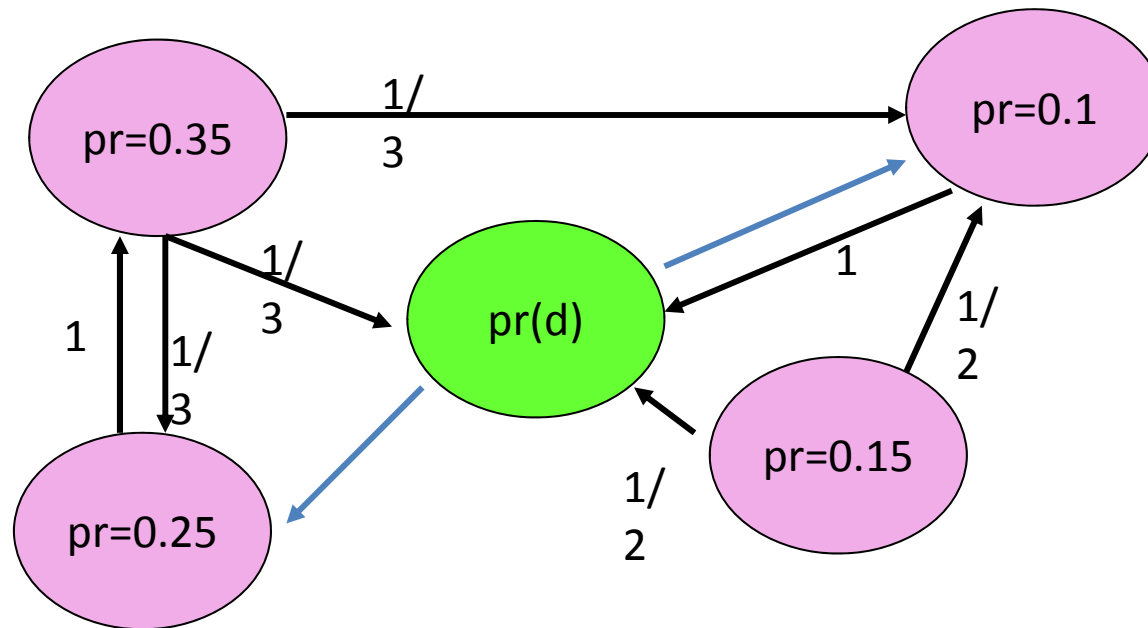
Simplified Page Rank Calculation

- Once $pr(d)$ is accepted as a measure of the importance of d there is a natural consequence;
- In the calculation of $pr(d)$, a hyperlink from a page d_1 to d should count for more than a hyperlink from page d_2 to d if $pr(d_1) > pr(d_2)$
- This motivates:

$$pr(d) = \sum_{e \in L(d)} pr(e)w_{ed}$$

where $L(d)$ is the set of pages which link to page d .

Simplified Page Rank Calculation



$$pr(d) = 0.35 * \frac{1}{3} + 0.1 * 1 + 0.15 * \frac{1}{2} = 0.292$$

Simplified Page Rank Calculation

- Change $pr(d)$ will change the page ranks of other pages, which in turn change $pr(d)$;
- In other words, the definition of page rank is recursive,

$$pr_{n+1}(d) = \sum_{e \in L(d)} pr_n(e) w_{ed}$$

Markov Chain Interpretation

- Let

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1D} \\ w_{21} & w_{22} & \cdots & w_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ w_{D1} & w_{D2} & \cdots & w_{1D} \end{pmatrix}$$

- Notice that each row of W sums to 1.

Markov Chain Interpretation

$$\begin{bmatrix} pr_{n+1}(1) \\ pr_{n+1}(2) \\ \dots \\ pr_{n+1}(D) \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & \dots & w_{D1} \\ w_{12} & w_{22} & \dots & w_{D2} \\ \dots & \dots & \dots & \dots \\ w_{1D} & w_{2D} & \dots & w_{DD} \end{bmatrix} \begin{bmatrix} pr_n(1) \\ pr_n(2) \\ \dots \\ pr_n(D) \end{bmatrix}$$

- If the system converges, then $pr = W^T pr$.
- pr is the invariant distribution of the Markov chain.
- In other word, pr is an eigenvector of W^T .

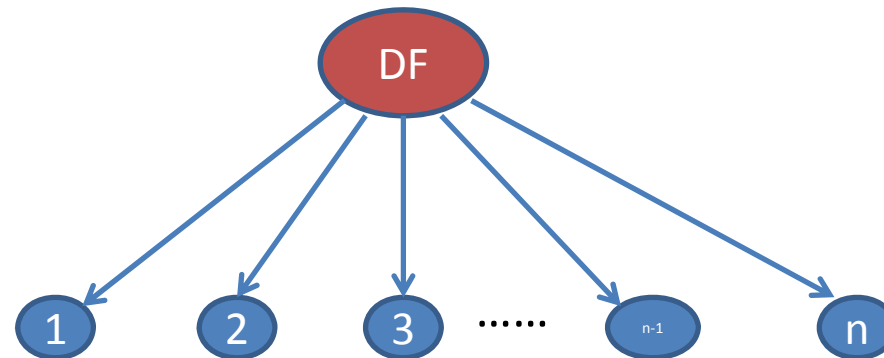
Damping Factor

- It represents the change that user stopping clicking links and get bored with the current page and then request another random page.
- If the damping factor is 85% then there is assumed to be about a 15% chance that a typical users won't follow any links on the page and instead navigate to a new random URL

Page Rank

- Taking into account the damping factor

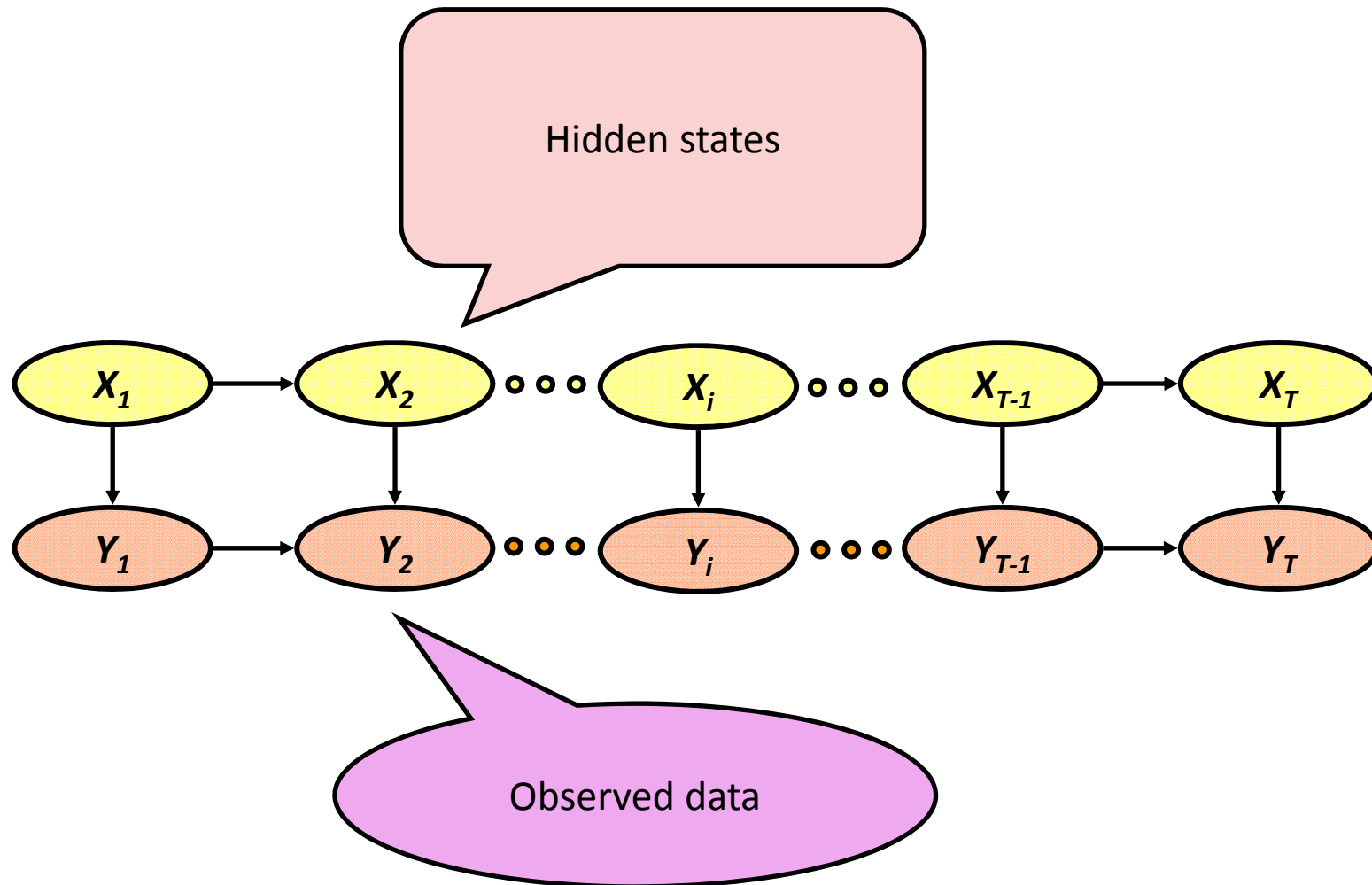
$$pr(d) = \frac{1 - \alpha}{N} + \alpha \left(\sum_{e \in L(d)} \frac{pr(e)}{l(e)} \right)$$



Part III

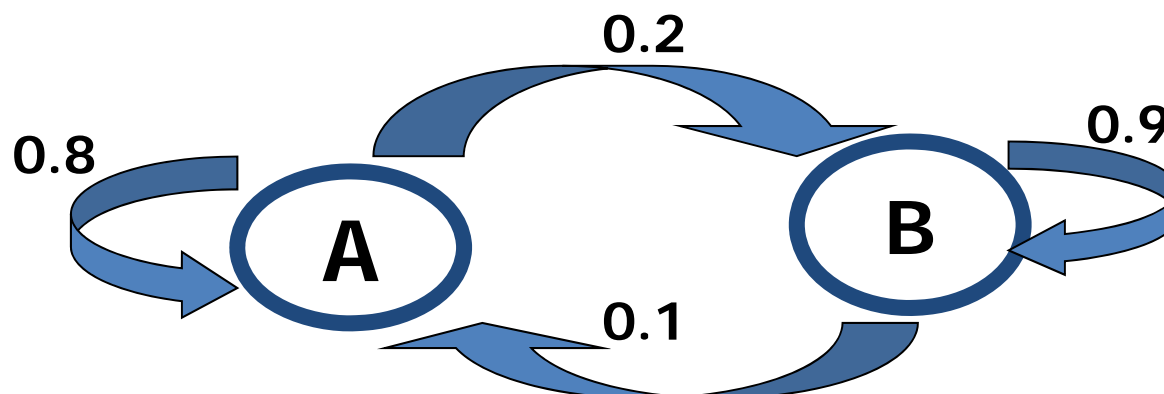
Hidden Markov Model

Hidden Markov Models - HMM



HMM—韦小宝的骰子

- 两种骰子，开始以 $2/5$ 的概率出千。
 - 正常A：以 $1/6$ 的概率出现每个点
 - 不正常B：5,6出现概率为 $3/10$,其它为 $1/10$
- 出千的随机规律



HMM例1—韦小宝的骰子

- 观测到其一次投掷结果

$$O = (1, 3, 4, 5, 5, 6, 6, 3, 2, 6)$$

- 问题：请判断韦小宝什么时候出千了？

隐马氏模型的数学模型

- 隐过程为 $X=\{X_1,\cdots,X_T\}$
- 观察过程为 $Y=\{Y_1,\cdots,Y_T\}$
- 模型参数 $\lambda = \{ \pi, \mathbf{A}, \mathbf{B} \}$
 - 初始分布 $\pi=(\pi_i)$, $\pi_i=P\{X_1=i\}$
 - 转移矩阵 $\mathbf{A}=(a_{ij})$, $a_{ij}=P(X_{n+1}=j \mid X_n=i)$
 - 给定某个时间的隐状态的情况下, 观测的分布矩阵 $\mathbf{B}=(b_{il})$, $b_{il}=P(Y_n=l \mid X_n=i)$ 。

隐马氏模型的数学问题

- 识别问题— 已知若干个隐马氏模型及其参数， 对一个观测样本， 决定它来自哪一个模型。
- 解码问题— 由观测样本得到隐状态；
- 学习问题— 由观测样本得到参数组 λ ；

Part IV

隐马氏模型(HMM)理论

识别问题

- 在已知若干个模型及其参数的情况下,识别问题就是一个对于给定样本进行 Bayesian 判决的问题。
- 判决步骤:
 - 根据参数求出在每一个模型中, 出现给定样本的概率 $P(Y | \lambda)$, 归一化就得到给定样本来自每个模型的概率 $P(\lambda | Y)$ 。
 - 利用 Bayesian 原理, 就可以得到最好模型猜测。

观测序列的概率计算

$$\begin{aligned} Pr(Y = y|\lambda) &= \sum_{X=x} Pr(Y = y|X = x, \lambda) Pr(X = x|\lambda) \\ &= \sum_{x=(x_1, \dots, x_T)} \pi(x_1) b_{x_1}(y_1) a_{x_1 x_2} b_{x_2}(y_2) \cdots a_{x_{T-1} x_T} b_{x_T}(y_T) \end{aligned}$$

枚举复杂度 $2TN^T$

多项式复杂度算法：前传算法和后传算法

前传概率

$$\alpha_t(i) = Pr(y_1, y_2, \cdots, y_t, x_t = i | \lambda)$$

$$\begin{aligned} a_{t+1}(i) &= Pr(y_1, y_2, \cdots, y_{t+1}, x_{t+1} = i | \lambda) \\ &= \sum_j Pr(y_1, y_2, \cdots, y_{t+1}, x_t = j, x_{t+1} = i | \lambda) \\ &= \sum_j Pr(y_1, y_2, \cdots, y_t, x_t = j | \lambda) Pr(y_{t+1}, x_{t+1} = i | x_t = j, \lambda) \\ &= \sum_j \alpha_t(j) a_{ji} b_i(y_{t+1}) \end{aligned}$$

前传算法 (Forward Algorithm)

- 初始化

$$\alpha_1(i) = \pi_i b_i(y_1), i = 1, 2, \dots, N.$$

- 迭代

$$\alpha_{t+1}(i) = \sum_{j=1}^N \alpha_t(j) a_{ji} b_i(y_{t+1})$$

$$i = 1, 2, \dots, N, \quad t = 1, 2, \dots, T - 1.$$

- 结果

$$Pr(Y | \lambda) = \sum_{i=1}^N \alpha_T(i).$$

后传概率

$$\beta_t(i) = Pr(y_{t+1}, y_{t+2}, \cdots, y_T | x_t = i, \lambda)$$

$$\begin{aligned}\beta_t(i) &= \sum_j Pr(y_{t+1}, y_{t+2}, \cdots, y_T, x_{t+1} = j | x_t = i, \lambda) \\ &= \sum_j Pr(y_{t+2}, \cdots, y_T | x_{t+1} = j, \lambda) Pr(y_{t+1}, x_{t+1} = j | x_t = i, \lambda) \\ &= \sum_j \beta_{t+1}(j) a_{ij} b_j(y_{t+1})\end{aligned}$$

后传算法(Backward Algorithm)

- 初始化

$$\beta_T(i) = 1, i = 1, 2, \dots, N;$$

- 迭代

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(y_{t+1})$$
$$1 \leq i \leq N, \quad t = T - 1, \dots, 1;$$

- 结果

$$Pr(Y|\lambda) = \sum_{i=1}^N \beta_1(i) \pi_i b_i(y_1).$$

解码问题(I)

- 问题： 给定观测序列 $Y = (y_1, y_2, \dots, y_T)$, 如何给出隐状态序列 $X^0 = (x_1^0, x_2^0, \dots, x_T^0)$.

- 单点最优

$$\gamma_t(i) = Pr(X_t = i|Y)$$

$$X'_t = \underset{i}{\operatorname{Argmax}} \gamma_t(i)$$

- 路径最优指： 对任意的 $X^0 = (x_1, x_2, \dots, x_T)$ 有

$$\begin{aligned} & Pr(x'_1, x'_2, \dots, x'_T | y_1, \dots, y_T) \\ & \geq Pr(x_1, x_2, \dots, x_T | y_1, \dots, y_T) \end{aligned}$$

解码问题(II)

- 由Bayesian公式有

$$\begin{aligned} & Pr(x'_1, x'_2, \dots, x'_T | y_1, \dots, y_T) \\ &= \frac{Pr(x_1, x_2, \dots, x_T, y_1, \dots, y_T)}{Pr(y_1, \dots, y_T)} \end{aligned}$$

- 又由于序列 \mathbf{Y} 给定, 问题等价于找最优的 \mathbf{x}^0 使联合概率 $Pr(x_1, \dots, x_T; y_1, \dots, y_T)$ 最大。

最优单点确定

$$\begin{aligned}\gamma_t(i) &= Pr(X_t = i | y_1, y_2, \dots, y_T, \lambda) \\ &= \frac{Pr(X_t = i, y_1, \dots, y_T | \lambda)}{Pr(y_1, \dots, y_T | \lambda)} \\ &= \frac{Pr(X_t = i, y_1, y_2, \dots, y_T | \lambda)}{\sum_i Pr(X_t = i, y_1, y_2, \dots, y_T | \lambda)} \\ &= \frac{Pr(y_{t+1}, \dots, y_T | x_t = i, y_1, \dots, y_t, \lambda) Pr(x_t = i, y_1, \dots, y_t | \lambda)}{\sum_i Pr(X_t = i, y_1, y_2, \dots, y_T | \lambda)} \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_i \alpha_t(i) \beta_t(i)}\end{aligned}$$

Viterbi算法(I)

- 算法的思想动态规划的递推算法。
- 递推变量为

$$\delta_t(i) = \max_{x_1, \dots, x_{t-1}} Pr(x_1, \dots, x_{t-1}, x_t = i, y_1, \dots, y_t | \lambda)$$

- 我们有递推公式

$$\begin{aligned} \delta_{t+1}(i) &= \max_{x_1, \dots, x_t} Pr(x_1, \dots, x_t, x_{t+1} = i, y_1, \dots, y_{t+1} | \lambda) \\ &= \left(\max_j \delta_t(j) a_{ji} \right) b_i(y_{t+1}) \end{aligned}$$

- 以 $\psi_t(i)$ 记录 t 时刻时使 $\delta_t(j)a_{ji}$ 最大的状态 j 。

Viterbi算法(II)

- 初始化

$$\begin{aligned}\delta_1(i) &= \pi_i b_i(y_1), \\ \psi_1(i) &= 0, \quad i = 1, 2, \dots, N.\end{aligned}$$

- 迭代

$$\begin{aligned}\delta_t(j) &= \left(\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} \right) b_j(y_t) \\ \psi_t(j) &= \underset{1 \leq i \leq N}{\operatorname{Argmax}} (\delta_{t-1}(i) a_{ij}) \\ t &= 2, \dots, T; \quad j = 1, \dots, N.\end{aligned}$$

Viterbi算法(III)

- 终止

$$p^* = \max_{1 \leq i \leq N} \delta_T(i)$$

$$x_T^* = \operatorname{Argmax}_{1 \leq i \leq N} \delta_T(i)$$

- 后推

$$x_t^* = \psi_{t+1}(x_{t+1}^*)$$

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

Viterbi算法实例(I)

- 转移概率以及初概率

	A	B
A	0.8	0.2
B	0.1	0.9
初概率	0.6	0.4

- 条件概率(Emission Probability)

	V_1	V_2	V_3	V_4	V_5	V_6
A	1/6	1/6	1/6	1/6	1/6	1/6
B	0.1	0.1	0.1	0.1	0.3	0.3

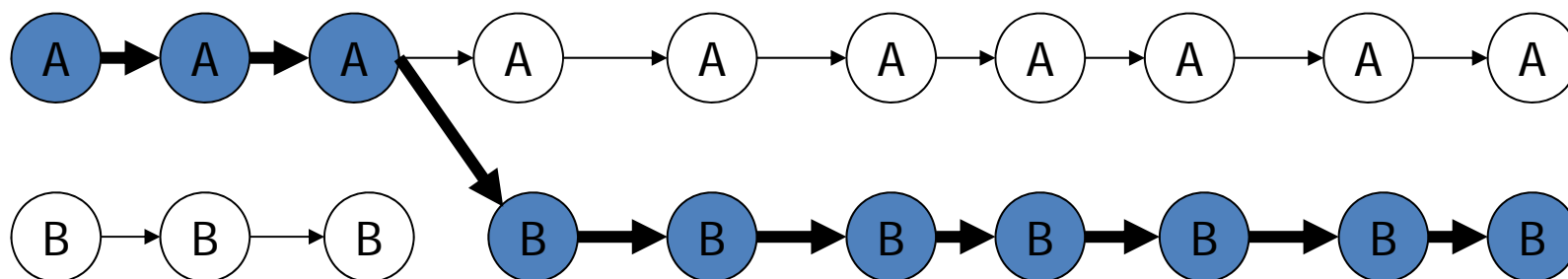
Viterbi算法实例(II)

	y_t	$\delta_t(A)$	$\psi_t(A)$	$\delta_t(B)$	$\psi_t(B)$
t=1	1	1.000×10^{-1}	-	4.000×10^{-2}	-
t=2	3	1.333×10^{-2}	A	3.600×10^{-3}	B
t=3	4	1.778×10^{-3}	A	3.240×10^{-4}	B
t=4	5	3.370×10^{-4}	A	1.067×10^{-4}	A
t=5	5	3.161×10^{-4}	A	2.880×10^{-5}	B
t=6	6	4.214×10^{-6}	A	7.776×10^{-6}	B
t=7	6	5.619×10^{-7}	A	2.100×10^{-6}	B
t=8	3	7.492×10^{-8}	A	1.890×10^{-7}	B
t=9	2	9.989×10^{-9}	A	1.701×10^{-8}	B
t=10	6	1.322×10^{-9}	A	4.592×10^{-9}	B

Viterbi算法实例(III)

观测序列为:

1 3 4 5 5 6 6 3 2 6



解码出来的状态序列为:

A A A B B B B B B B

HMM学习问题

- 学习问题:
 - 就是由观测估计模型参数.
- 学习的两种情况:
 - 观测链相应的状态链已知;
 - 观测链相应的状态链未知。

学习原则

- 极大似然估计(MLE)

$$\hat{\lambda} = \underset{\lambda}{\operatorname{Argmax}} Pr(y_1, \dots, y_T | \lambda)$$

- 状态链已知时

$$Pr(y_1, \dots, y_T, X_1, \dots, X_T | \lambda)$$

- 状态链未知时

$$\sum_{(X_1, \dots, X_T)} Pr(y_1, \dots, y_T, X_1, \dots, X_T | \lambda)$$

状态链已知时的MLE

$$\begin{aligned} & Pr(y_1, \cdots, y_T; X_1, \cdots, X_T | \lambda) \\ &= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}} \prod_{t=1}^T b_{X_t}(y_t) \\ &= \prod_i \pi_i^{1_i(X_1)} \prod_{i,j} a_{ij}^{\sum_{t=1}^{T-1} 1_i(X_t) 1_j(X_{t+1})} \prod_{i,l} b_i(l)^{\sum_{t=1}^T 1_i(X_t) 1_l(y_t)} \\ &= \prod_i \pi_i^{C_i} \prod_{i,j} a_{ij}^{A_{ij}} \prod_{i,l} b_i(l)^{B_{il}} \end{aligned}$$

简单优化问题

$$\text{Max: } \sum_k z_k \log x_k$$

$$\text{subject to: } \sum_k x_k = 1$$

$$\text{Estimation: } x_i = \frac{z_i}{\sum_k z_k}, i = 1, \dots, N.$$

参数估计(状态已知)

- 把从状态 i 到转移为状态 j 的频数记为 A_{ij} , 可估计转移概率 a_{ij} , 为

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^N A_{ij}}$$

- 同样记状态 i 下观察到符号 s 的频数记为 B_{is} , 则可估计 b_{is} , 为

$$\hat{b}_{is} = \frac{B_{is}}{\sum_{j=1}^M B_{ij}}$$

参数估计评价

- 隐Markov模型的状态链要有充分长的样本(大数定律, 以频率代替概率)。
- 不幸的是状态链往往并不知道, 而只是可以得到估计, 不修正地使用频率估计会增加误差, 且这种估计不稳健。

参数估计的EM思想

- 当状态链未知时，由于似然函数的计算中包含了对所有可能的状态链的求和，计算过大，在实际中是不可能被采用的。为此，人们采取折衷的方案，构造一个递推算法，使之能相当合理地给出模型参数的粗略估计。
- 其核心思想是：在当前参数下，用期望值当成频数“数数”，并用的“频率”估计概率。这实际上是一种EM迭代算法思想。

EM算法

- 实际上是 **E** (期望) 与 **M** (最大化) 两个步骤合起来构成的算法，称为**EM**算法.
- EM算法是针对测量数据不完全时, 求参数的最大似然估计的统计方法。
- HMM 的模型参数的估计, 是EM算法的一个最常见且极有用的一种典型例子.

EM算法基本框架

- 观测数据Y
- 缺失数据 X
- 完全数据 $Z=(Y, X)$.
- E-Step (取期望).

$$\hat{Z} = E(Z | Y, \theta^{(t-1)})$$

- M-step (取极大).

$$\theta^{(t)} = \underset{\theta}{\operatorname{Argmax}} L(\theta | \hat{Z}, \theta^{(t-1)})$$

期望频数(状态未知)

$$\begin{aligned}\xi_t(i, j) &= Pr(X_t = i, X_{t+1} = j | y_1, \dots, y_T, \lambda) \\&= \frac{Pr(X_t = i, X_{t+1} = j, y_1, \dots, y_T | \lambda)}{Pr(y_1, \dots, y_T | \lambda)} \\&= \frac{Pr(X_t = i, X_{t+1} = j, y_1, \dots, y_T | \lambda)}{\sum_i \sum_j Pr(X_t = i, X_{t+1} = j, y_1, \dots, y_T | \lambda)} \\&= \frac{Pr(x_{t+1} = j, y_{t+1}, \dots, y_T | x_t, y_1, \dots, y_t, \lambda) Pr(x_t = i, y_1, \dots, y_t | \lambda)}{Pr(y_1, \dots, y_T | \lambda)} \\&= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)} \\&= \frac{\alpha_t(i) a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{\sum_i \alpha_t(i) \beta_t(i)}\end{aligned}$$

期望频数(状态未知)

$$\begin{aligned}\gamma_t(i) &= Pr(x_t = i | y_1, \dots, y_T, \lambda) \\ &= \sum_{j=1}^N Pr(x_t = i, x_{t+1} = j | y_1, \dots, y_T, \lambda) \\ &= \sum_{j=1}^N \xi_t(i, j)\end{aligned}$$

Baum-Welch公式

$$\left\{ \begin{array}{l} \bar{\pi}_i = r_1(i) \\ \bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ \bar{b}_j(l) = \frac{\sum_{t=1}^T \gamma_t(j) \delta(y_t, V_l)}{\sum_{t=1}^T \sum_j \gamma_t(j)} \end{array} \right.$$

Baum-Welch公式的推导(1)

- 我们定义一个描述模型“趋势”的量，以衡量参数估计前后的概率分布的差异。相对熵是最好的选择

$$Q(\bar{\lambda}|\lambda) = \sum_X P(X, Y|\lambda) \log Pr(X, Y|\bar{\lambda})$$

Baum-Welch公式的推导(2)

$$\begin{aligned} & Q(\bar{\lambda}|\lambda) - Q(\lambda|\lambda) \\ &= \sum_X Pr(X, Y|\lambda) \log \frac{Pr(X, Y|\bar{\lambda})}{Pr(X, Y|\lambda)} \\ &\leq \sum_X Pr(X, Y|\lambda) \left(\frac{Pr(X, Y|\bar{\lambda})}{Pr(X, Y|\lambda)} - 1 \right) \\ &= Pr(Y|\bar{\lambda}) - Pr(Y|\lambda) \end{aligned}$$

- 说明只要依据Q函数增大来更新参数，就能够使得似然函数朝变大的方向改进，而且一个观测Y对应了一次改进。

Baum-Welch公式的推导(3)

- 于是要想得到参数修改的递推公式, 只要把模型 λ_m 修改为更好的模型 λ_{m+1} , 即只需将它取得使下式成立,

$$\lambda_{m+1} = \underset{\lambda}{\operatorname{Argmax}} Q(\lambda | \lambda_m)$$

Baum-Welch公式的推导(4)

- 写出Q函数的表达式，我们有

$$\begin{aligned} Q(\bar{\lambda}|\lambda) &= \sum_X Pr(X, Y|\lambda) \log Pr(X, Y|\bar{\lambda}) \\ &= \sum_X Pr(X, Y|\lambda) \left(\log \bar{\pi}_{x_1} + \sum_{t=1}^{T-1} \log \bar{a}_{x_t x_{t+1}} + \sum_{t=1}^T \log \bar{b}_{x_t}(y_t) \right) \\ &= \sum_i \left(\sum_X Pr(X, Y|\lambda) 1_i(x_1) \right) \log \bar{\pi}_i \\ &\quad + \sum_i \sum_j \left(\sum_{t=1}^{T-1} \sum_X Pr(X, Y|\lambda) 1_i(x_t) 1_j(x_{t+1}) \right) \log \bar{a}_{ij} \\ &\quad + \sum_i \sum_l \left(\sum_{t=1}^T \sum_X Pr(X, Y|\lambda) 1_i(x_t) 1_l(y_t) \right) \log \bar{b}_i(l) \end{aligned}$$

Baum-Welch公式的推导(5)

$$\begin{aligned} &= \sum_i (Pr(x_1 = i, Y|\lambda)) \log \bar{\pi}_i \\ &+ \sum_i \sum_j \left(\sum_{t=1}^{T-1} Pr(x_t = i, x_{t+1} = j, Y|\lambda) \right) \log \bar{a}_{ij} \\ &+ \sum_i \sum_l \left(\sum_{t=1}^T \sum_X Pr(x_t = i, Y|\lambda) 1_l(y_t) \right) \log \bar{b}_i(l) \end{aligned}$$

- 对每个变量可以分别取最大值。

初概率的重估计

$$\text{Max: } Q_{\pi}(\bar{\pi}|\lambda) = \sum_i P(x_1 = i, Y|\lambda) \log \bar{\pi}_i$$

$$\text{Subject to: } \sum_i \bar{\pi}_i = 1$$

$$\pi_i^{(m+1)} = \frac{P(x_1 = i, Y|\lambda^{(m)})}{P(Y|\lambda^{(m)})} = \gamma_1(i)$$

转移概率重估计

$$\text{Max: } Q_{a_i}(\overline{a_i}|\lambda) = \sum_j \left(\sum_{t=1}^{T-1} P(x_t = i, x_{t+1} = j, Y|\lambda) \right) \log \overline{a_{ij}}$$

$$\text{Subject to: } \sum_j \overline{a_{ij}} = 1$$

$$a_{ij}^{(m+1)} = \frac{\sum_{t=1}^{T-1} P(x_t = i, x_{t+1} = j, Y|\lambda^{(m)})}{\sum_{t=1}^{T-1} P(x_t = i, Y|\lambda^{(m)})}$$

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

观测概率重估计

$$\text{Max: } Q_{b_i}(\bar{b}_i|\lambda) = \sum_k \left(\sum_{t=1}^T P(x_t = i, Y|\lambda) 1_k(y_t) \right) \log \bar{b}_{ik}$$

$$\text{Subject to: } \sum_k \bar{b}_{ik} = 1$$

$$\begin{aligned} b_{ik}^{(m+1)} &= \frac{\sum_{t=1}^T \left(P(x_t = i, Y|\lambda^{(m)}) \delta(y_t, k) \right)}{\sum_{t=1}^T P(x_t = i, Y|\lambda^{(m)})} \\ &= \frac{\sum_{t=1}^T \gamma_t(i) \delta(y_t, k)}{\sum_{t=1}^T \gamma_t(i)} \end{aligned}$$

几点说明

- 更详细的内容可参见《应用随机过程》第十章 HMM.
- 在上面所述的算法中, 初始值 λ_0 的设置会直接影响到估计的好坏. 为此, 常用的一种方法是, 根据先验知识设置一条较长的“标准虚拟”状态链, 再用前面讲的已知观测链相应的状态链的情况下参数估计的方法, 得到一个对参数的粗估计, 并以它作为 λ_0 的取值。

Part V

隐马氏模型应用

应用1： 基因序列CpG岛识别

本部分ppt修改自网上资料

http://people.brandeis.edu/~moshep/Projects/ProjectPresentations2003_2/HMM/CpG_Islands_HMM_final.ppt

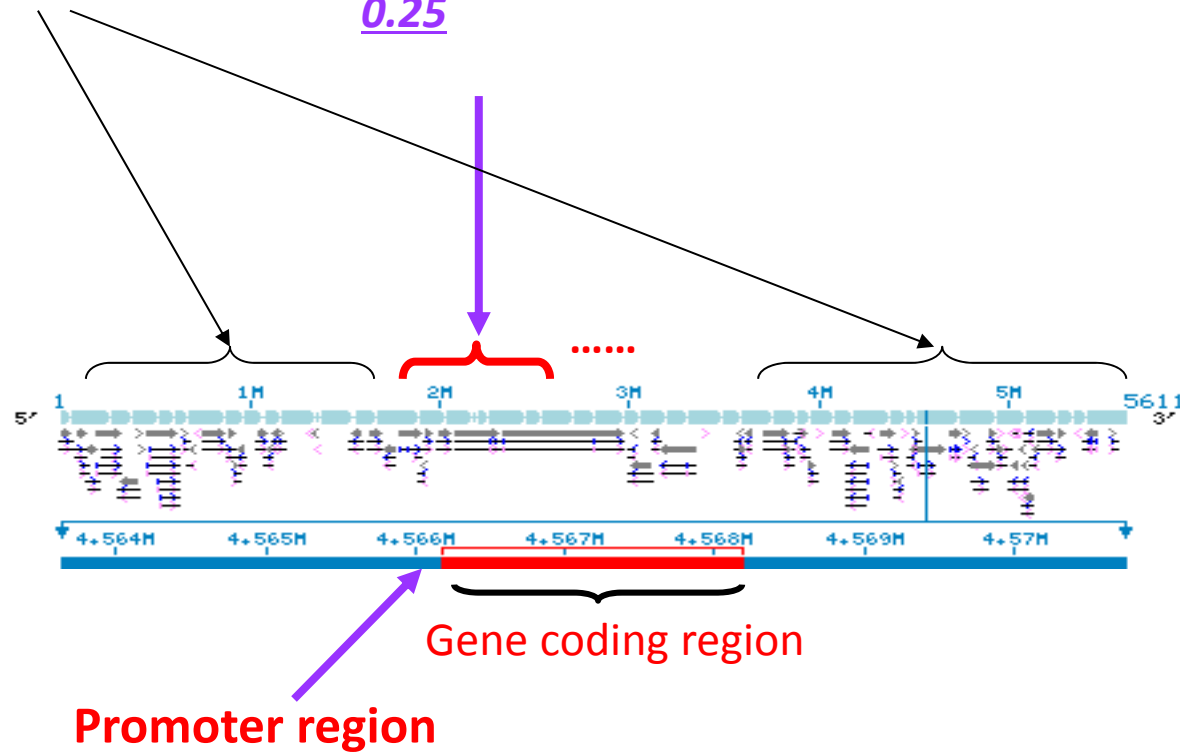
什么是CpG岛？

CG-poor regions: $P(CG)$

~ 0.07!

CG-rich region: $P(CG) \sim$

0.25



CpG 島

- Away from gene regions:
 - The C in CG pairs is usually *methyated*
 - *Methylation* inhibits gene transcription
 - These CGs tend to **mutate to TG**
- Near promoter and coding regions:
 - Methylation is suppressed:
 - ***CGs remain CGs***
 - Makes transcription easier!

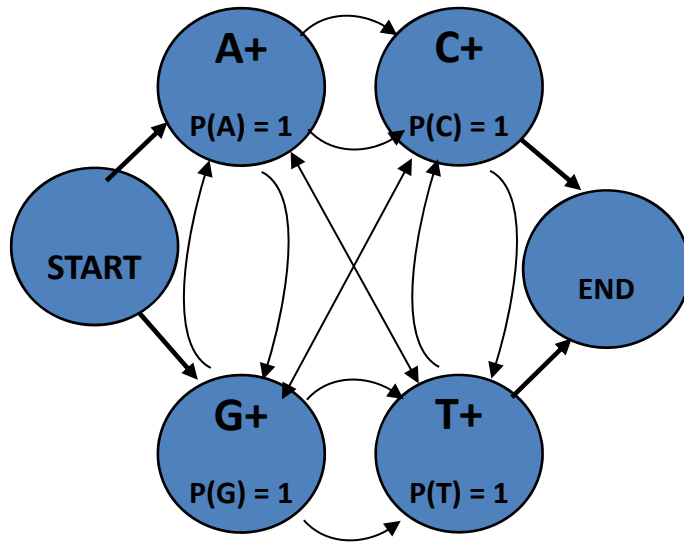
CpG岛的生物学意义

- CpG-rich regions are associated with genes which are *frequently transcribed*.
- Helps to understand gene expression related to *location* in genome.

HMM对于CpG岛识别的意义

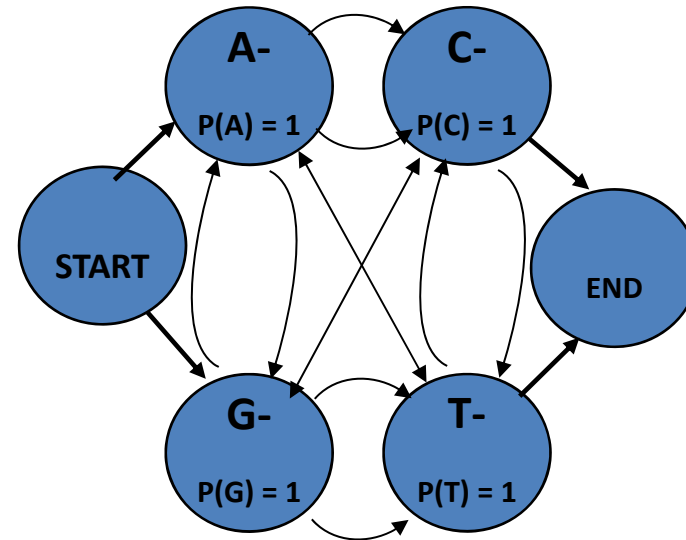
- Q: Why an HMM?
- It can answer the questions:
 - Short sequence: *does it come from a CpG island or not?*
 - Long sequence: *where are the CpG islands?*
- So, what's a good model?
 - Well, we need states for **ISLAND bases** and **NON-ISLAND bases ...**

HMM模型框架

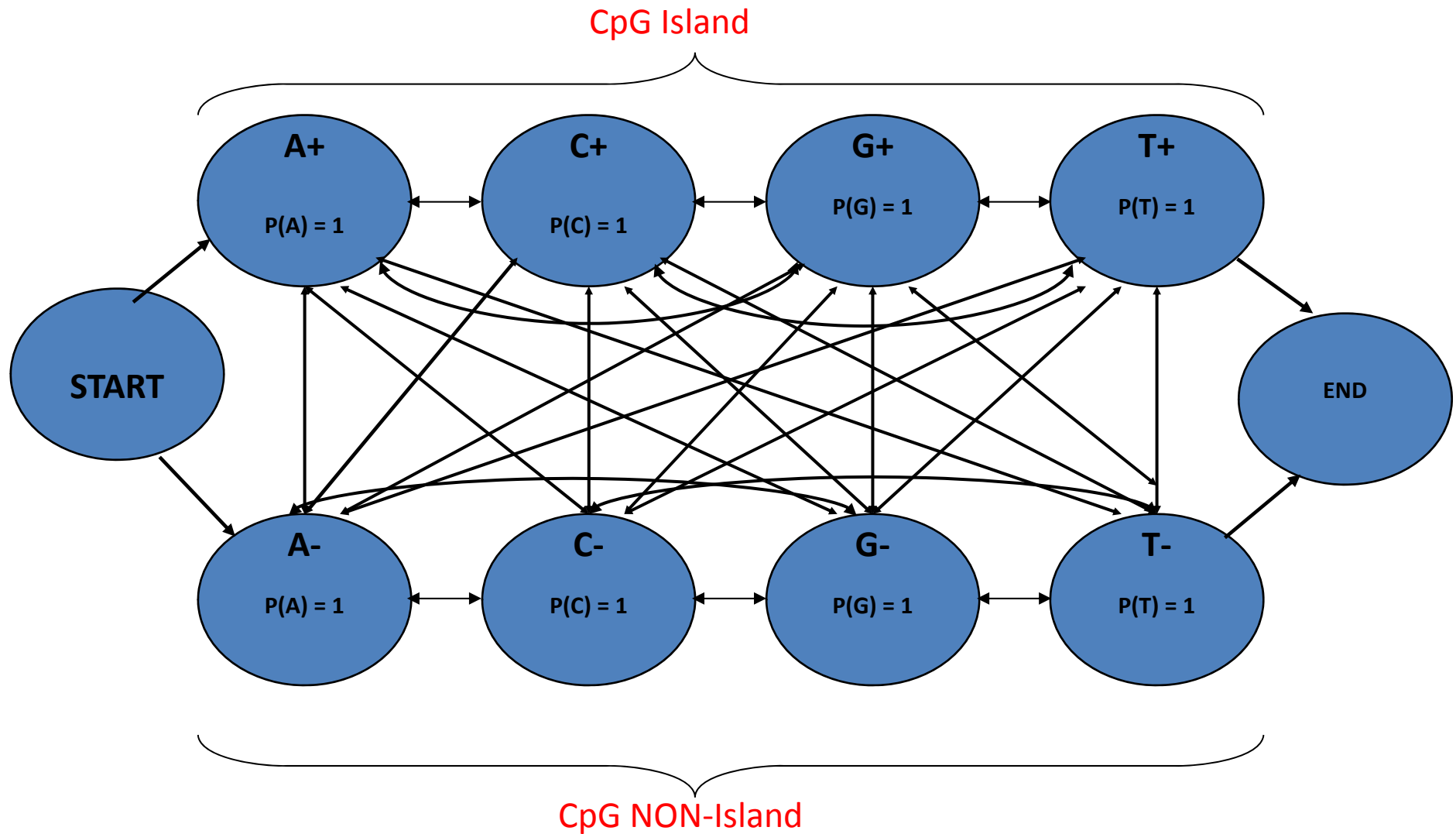


CpG Island (+)

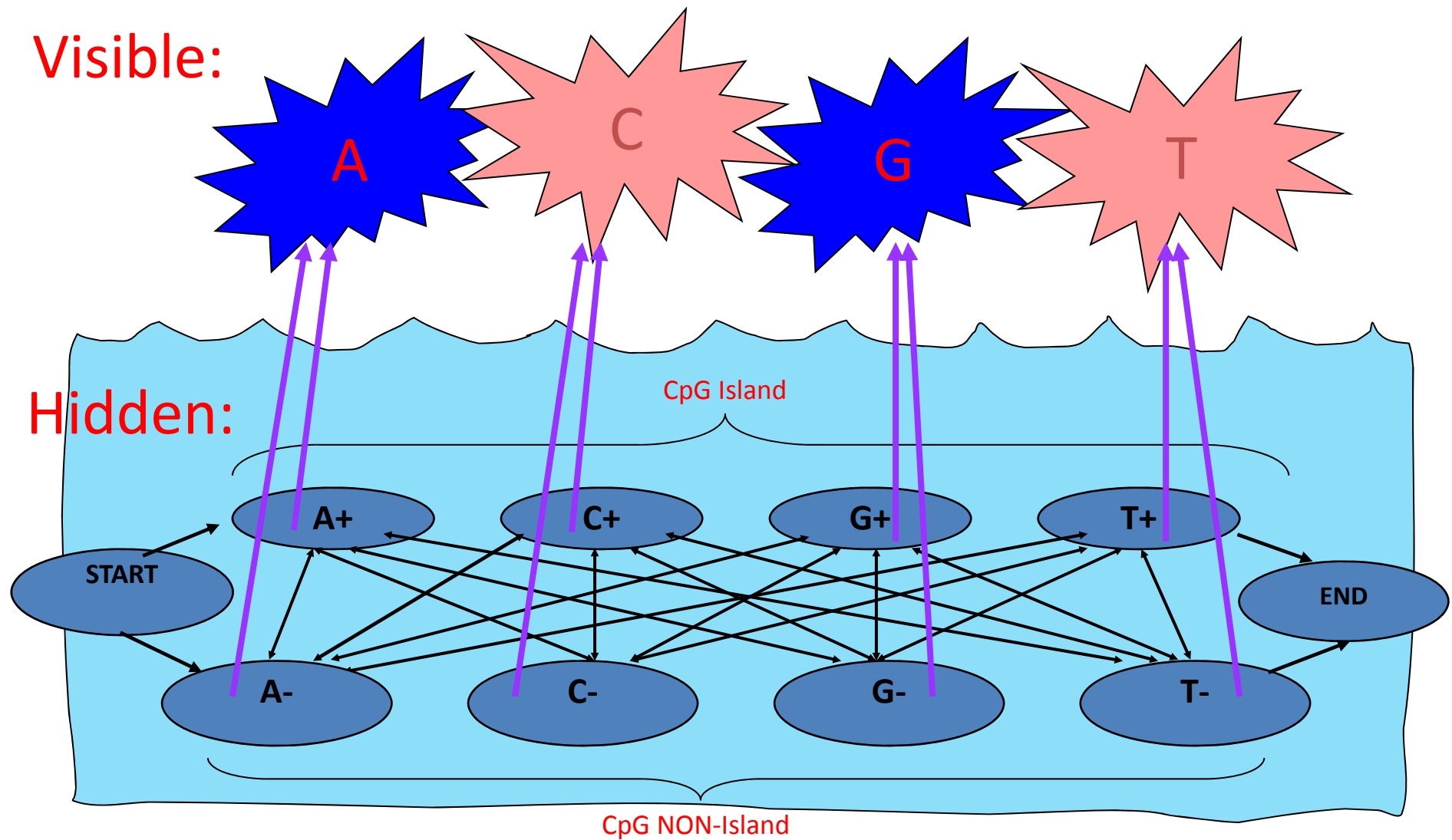
CpG NON-Island (-)



HMM模型框架



HMM示意



HMM训练问题

CG-RICH sequences

AATAGAGAGGTTTCGACTCTG
CATTTCCCAAATACGTAATGCT
TACGGTACACGACCCAAGCTC
TCTGCTTGAATCCCAAATCTG
AGCGGACAGATGAGGGGGC
GCAGAGGAAAAACAGGTTTT
GGACCCTACATAAANAGAGA
GGTTCGTAAATAGAGA

HOW?

ML or
Forward/
Backward
algorithm

CG-POOR sequences

GGTTCGACTCTGCATTTCCTCA
AATACGTAATGCTTACGGTTA
AATAGAGAGGTTTCGACTCTG
CATTTCCCAAATACGTAATGCT
TACGGTACACGACCCAAGCTC
TCTGCTTGTAACTTGTCTTNG
TCGCAGCTGGTCTTGCTTTG
CTGGGGCTGCTGA

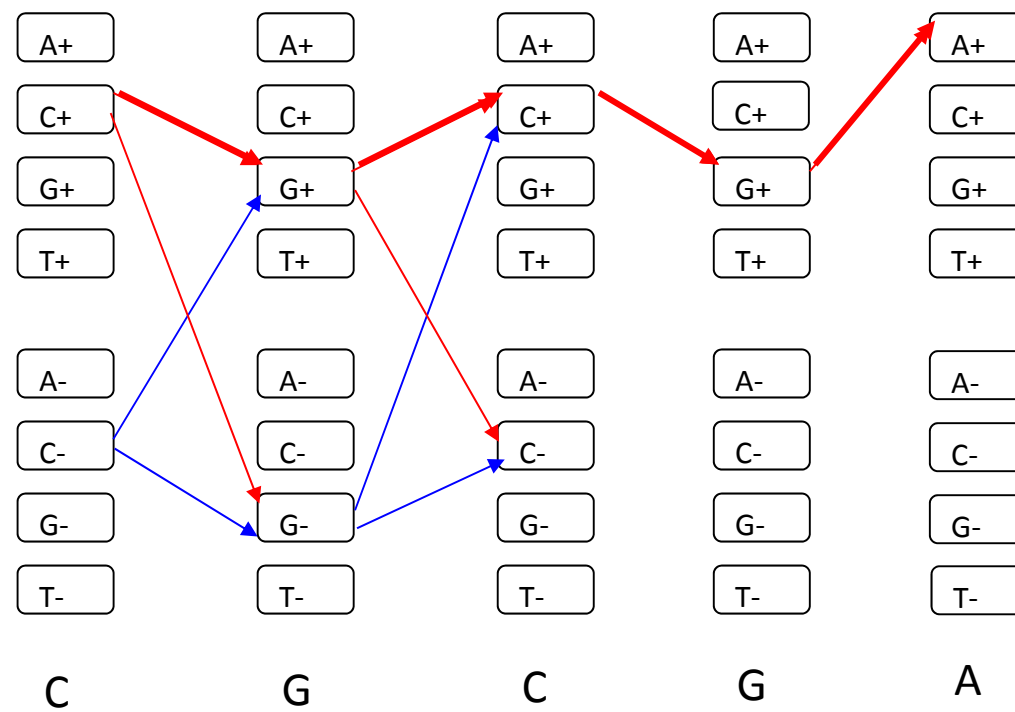
	A+	C+	G+	T+	A-	C-	G-	T-
A+	0.17	0.26	0.42	0.11	0.01	0.01	0.01	0.01
C+	0.16	0.36	0.26	0.18	0.01	0.01	0.01	0.01
G+	0.15	0.33	0.37	0.11	0.01	0.01	0.01	0.01
T+	0.07	0.35	0.37	0.17	0.01	0.01	0.01	0.01
A-	0.01	0.01	0.01	0.01	0.29	0.2	0.27	0.2
C-	0.01	0.01	0.01	0.01	0.31	0.29	0.07	0.29
G-	0.01	0.01	0.01	0.01	0.24	0.23	0.29	0.2
T-	0.01	0.01	0.01	0.01	0.17	0.23	0.28	0.28

HMM解码问题

Viterbi Algorithm

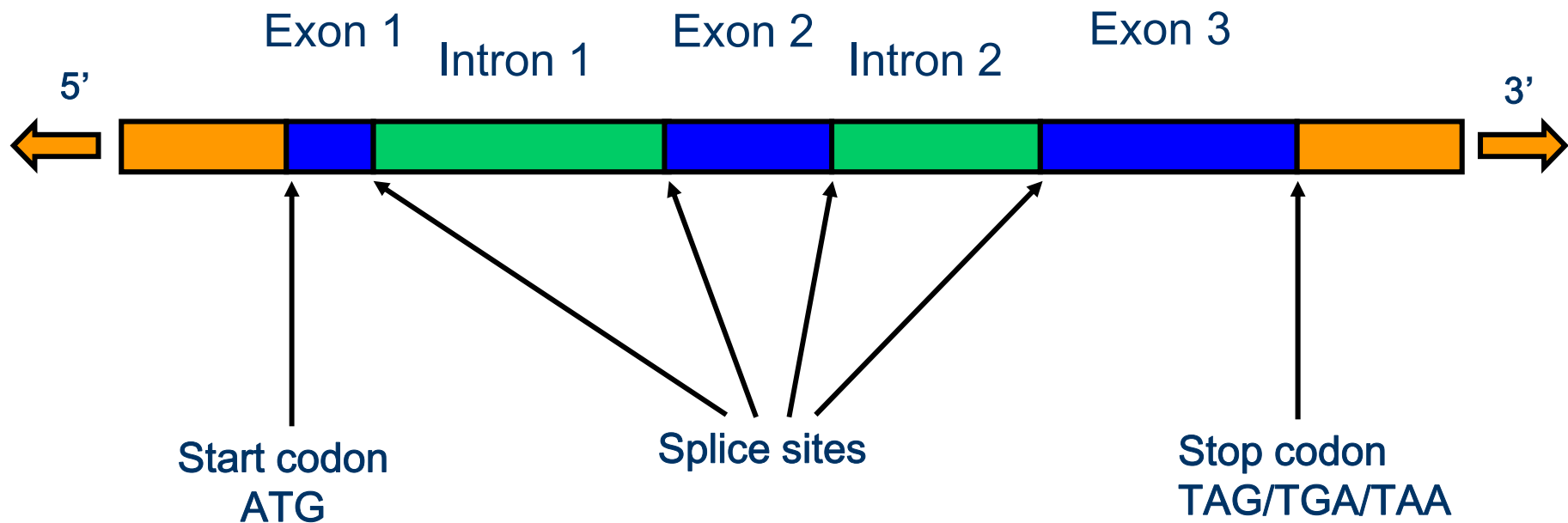
- Decoding- Meaning of observation sequence by looking at the underlying states.
- Hidden states A+,C+,G+,T+,A-,C-,G-,T-
- Observation sequence **CGCGA**
- State sequences C+,G+,C+,G+,A+ or C-,G-,C-,G-,A-
or C+,G-,C+,G-,A+
- Most Probable Path C+,G+,C+,G+,A+

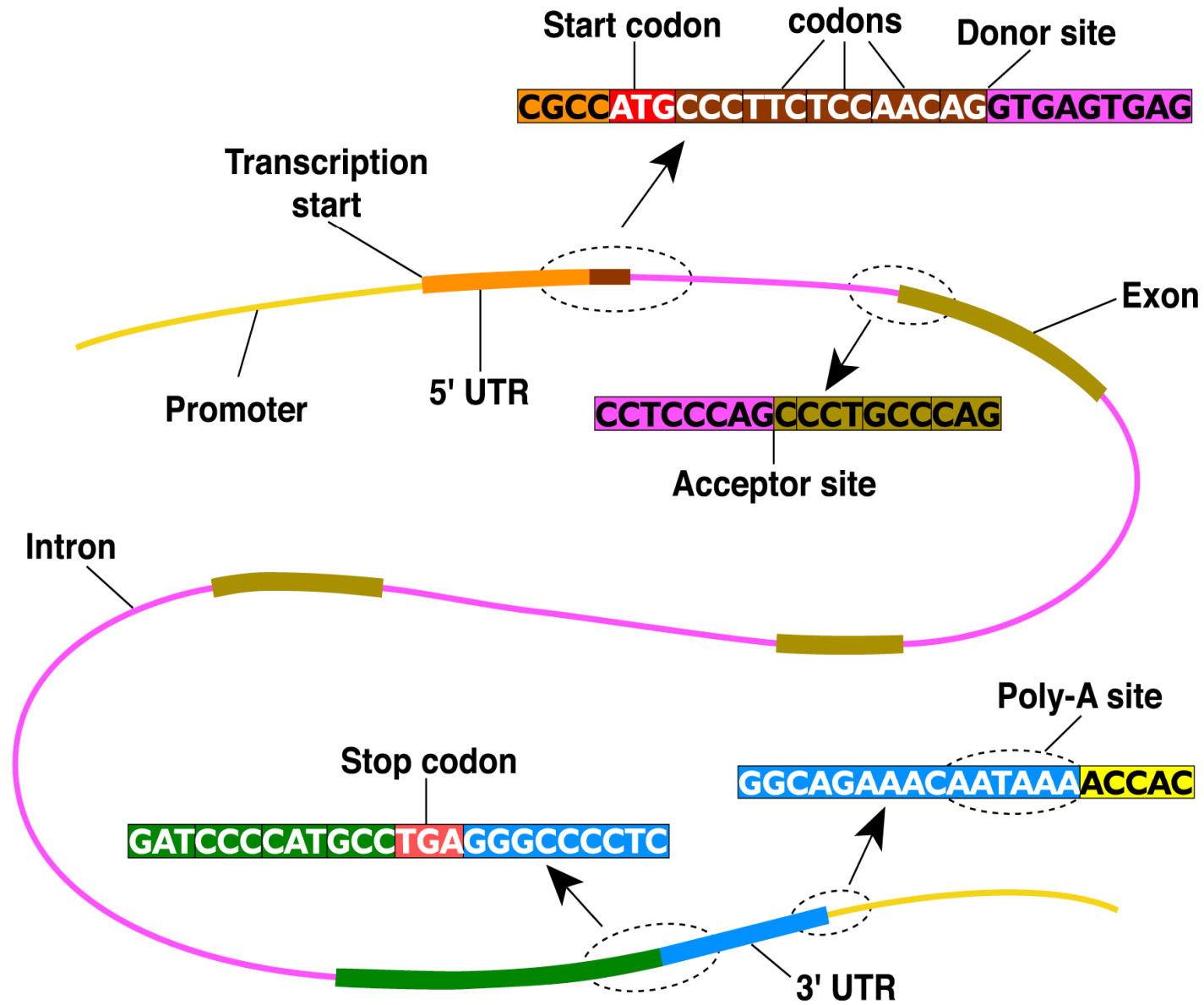
HMM解码问题



应用II: Gene Finding

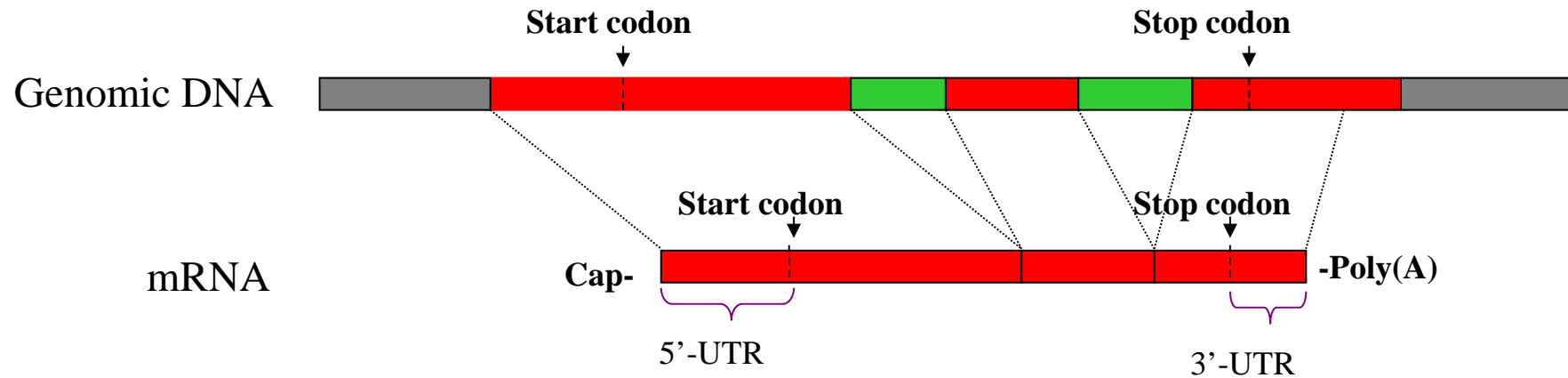
- This part is modified from slides download from www.cs.ubc.ca/~rogic/GeneFinding.ppt*





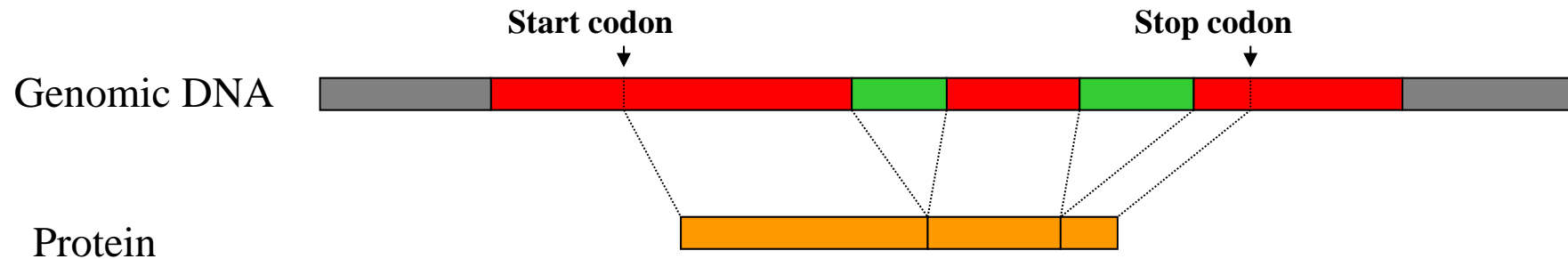
Spliced Alignment

Compare with cDNA or EST probes

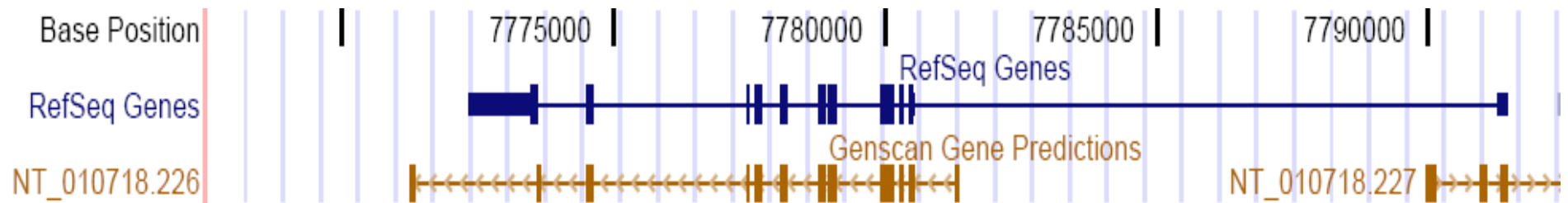


Spliced Alignment

Compare with Protein Probes

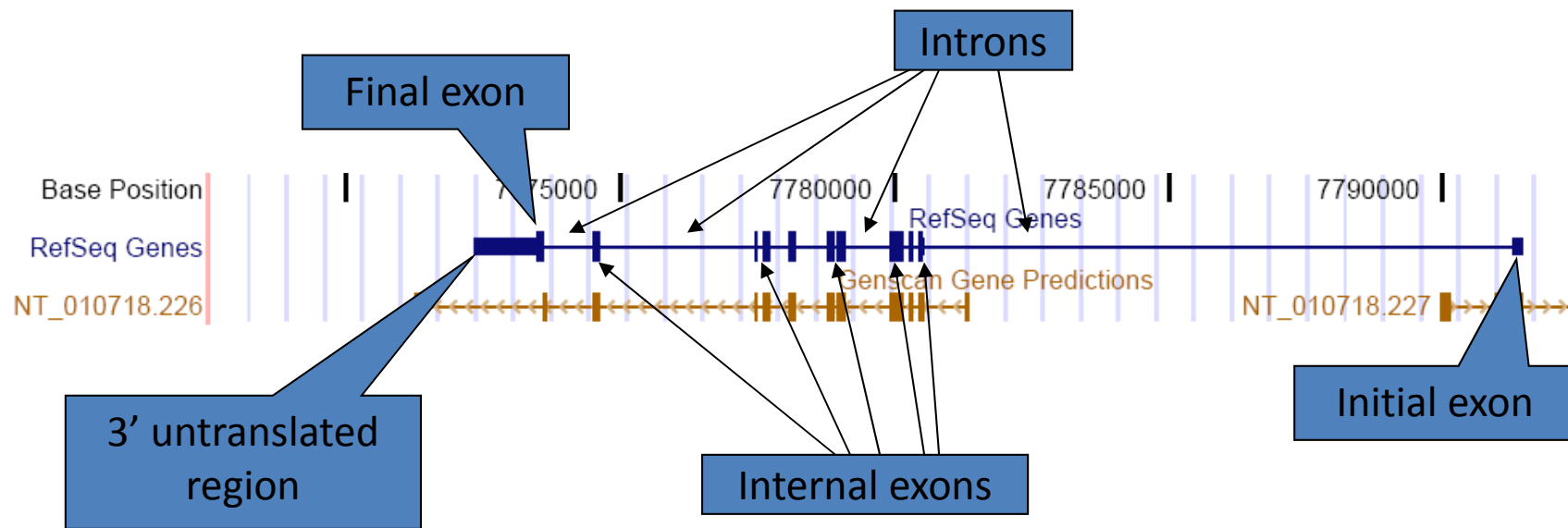


A Eukaryotic Gene



- This is the human p53 tumor suppressor gene on chromosome 17.
- Genscan is one of the most popular gene prediction algorithms.

A Eukaryotic Gene



This particular gene lies on the reverse strand.

An Intron

revcomp(CT)=AG

GT: signals **start** of intron

AG: signals **end** of intron

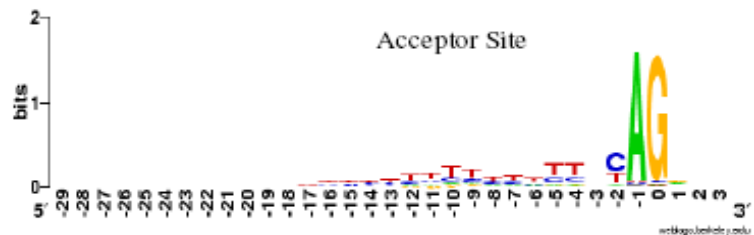
revcomp(AC)=GT



3' splice site



5' splice site



Signals vs Contents

- In gene finding, a small pattern within the genomic DNA is referred to as a **signal**, whereas a region of genomic DNA is a **content**.
- Examples of **signals**: splice sites, starts and ends of transcription or translation, branch points, transcription factor binding sites
- Examples of **contents**: exons, introns, UTRs, promoter regions

Prior Knowledge

- The translated region must have a length that is a multiple of 3.
- Some codons are more common than others.
- Exons are usually shorter than introns.
- The translated region begins with a start signal and ends with a stop codon.
- 5' splice sites (**exon to intron**) are usually GT;
- 3' splice sites (**intron to exon**) are usually AG.
- The distribution of nucleotides and dinucleotides is usually different in introns and exons.

Prior Knowledge

- We want to build a probabilistic model of a gene that incorporates our prior knowledge.
- E.g., the translated region must have a length that is a multiple of 3.

Prokaryotic Vs. Eukaryotic Gene Finding

Prokaryotes:

- small genomes $0.5 - 10 \cdot 10^6$ bp
- high coding density (>90%)
- no introns



- Gene identification relatively easy, with success rate $\sim 99\%$

Problems:

- overlapping ORFs
- short genes
- finding TSS and promoters

Eukaryotes:

- large genomes $10^7 - 10^{10}$ bp
- low coding density (<50%)
- intron/exon structure



- Gene identification a complex problem, gene level accuracy $\sim 50\%$

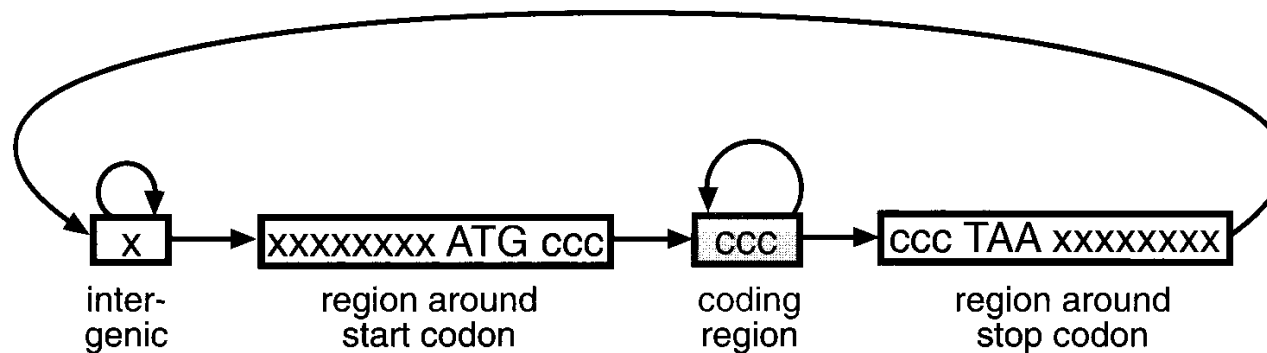
Problems:

- many

HMMs and Gene Structure

- Nucleotides $\{A, C, G, T\}$ are the observables
- Different states generate nucleotides at different frequencies

A simple HMM for unspliced genes:



AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

- The sequence of states is an annotation of the generated string – each nucleotide is generated in intergenic, start/stop, coding state

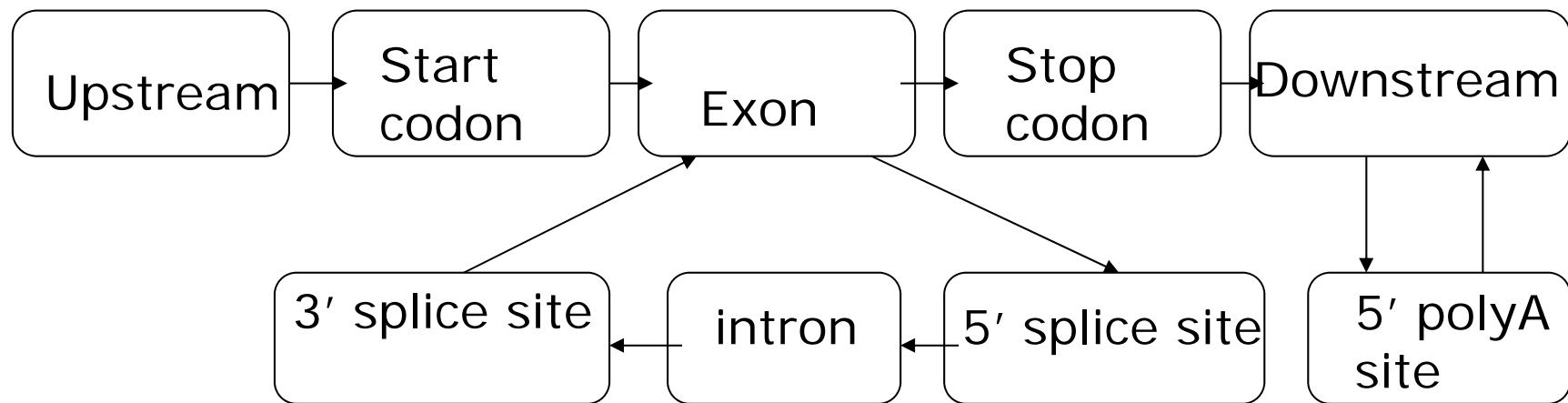
Examples of Gene Finders Using HMM

- [GeneMark](#) – HMMs enhanced with ribosomal binding site recognition
- [Genie](#) – neural networks for splicing, HMMs for coding sensors, overall structure modeled by HMM
- [Genscan](#) – Weight Matrix, Weight Array and decision trees as signal sensors, HMMs for content sensors, overall HMM
- [HMMgene](#) – HMM trained using conditional maximum likelihood
- [Morgan](#) – decision trees for exon classification, also Markov Models
- [VEIL](#) – sub-HMMs each to describe a different bit of the sequence, overall HMM

EXAMPLE: Finding Genes with VEIL

- The **Viterbi Exon-Intron Locator (VEIL)** was developed by John Henderson, Steven Salzberg, and Ken Fasman at Johns Hopkins University.
- Gene finder with a modular structure:
- Uses a HMM which is made up of sub-HMMs each to describe a different bit of the sequence: upstream noncoding DNA, exon, intron, ...
- Uses biological knowledge to “hardwire” part of HMM, eg. start + stop codons, splice sites.

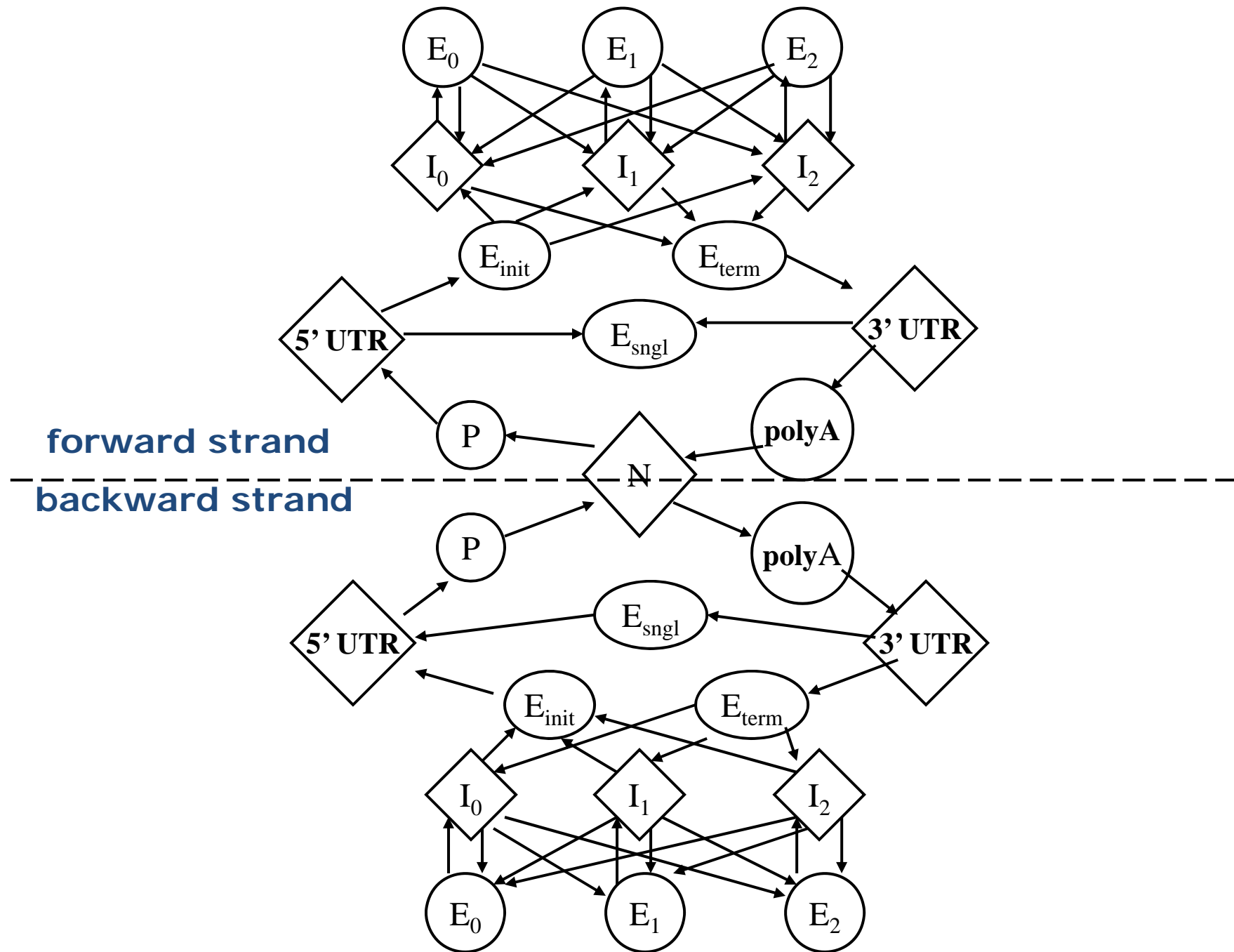
The Overall Model of VEIL



For more details, see J. Henderson, S.L. Salzberg, and K. Fasman (1997) *Journal of Computational Biology* 4:2, 127-141.

Genscan

- Developed by Chris Burge 1997
- One of the most accurate *ab initio* programs
- Uses explicit state duration HMM (semi-HMM) to model gene structure (different length distributions for exons)
- Different model parameters for regions with different GC content



GeneScan

- N - intergenic region
- P - promoter
- (5'/3') UTR - 5'/3' untranslated region
- E_{sngl} – single exon (intronless) (translation start -> stop codon)
- E_{init} – initial exon (translation start -> donor splice site)
- E_k – phase k internal exon (acceptor splice site -> donor splice site)
- E_{term} – terminal exon (acceptor splice site -> stop codon)
- I_k – phase k intron: 0 – between codons; 1 – after the first base of a codon; 2 – after the second base of a codon

GeneScan Features

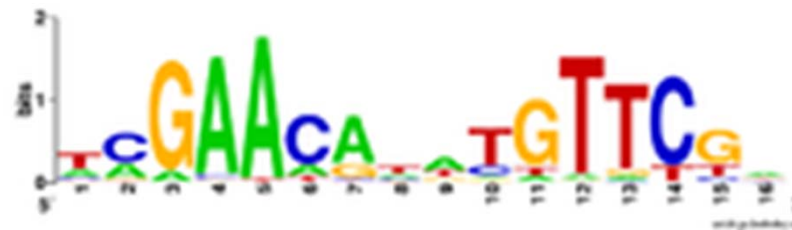
- Model both strands at once
- Each state may output a string of symbols (according to some probability distribution).
- Explicit intron/exon length modeling
- Advanced splice site modeling
- Parameters learned from annotated genes
- Prediction of multiple genes in a sequence (partial or complete).

PWM

- Position Weight matrix (PWM) for acceptor site, branch point, polyA site and promoter region.

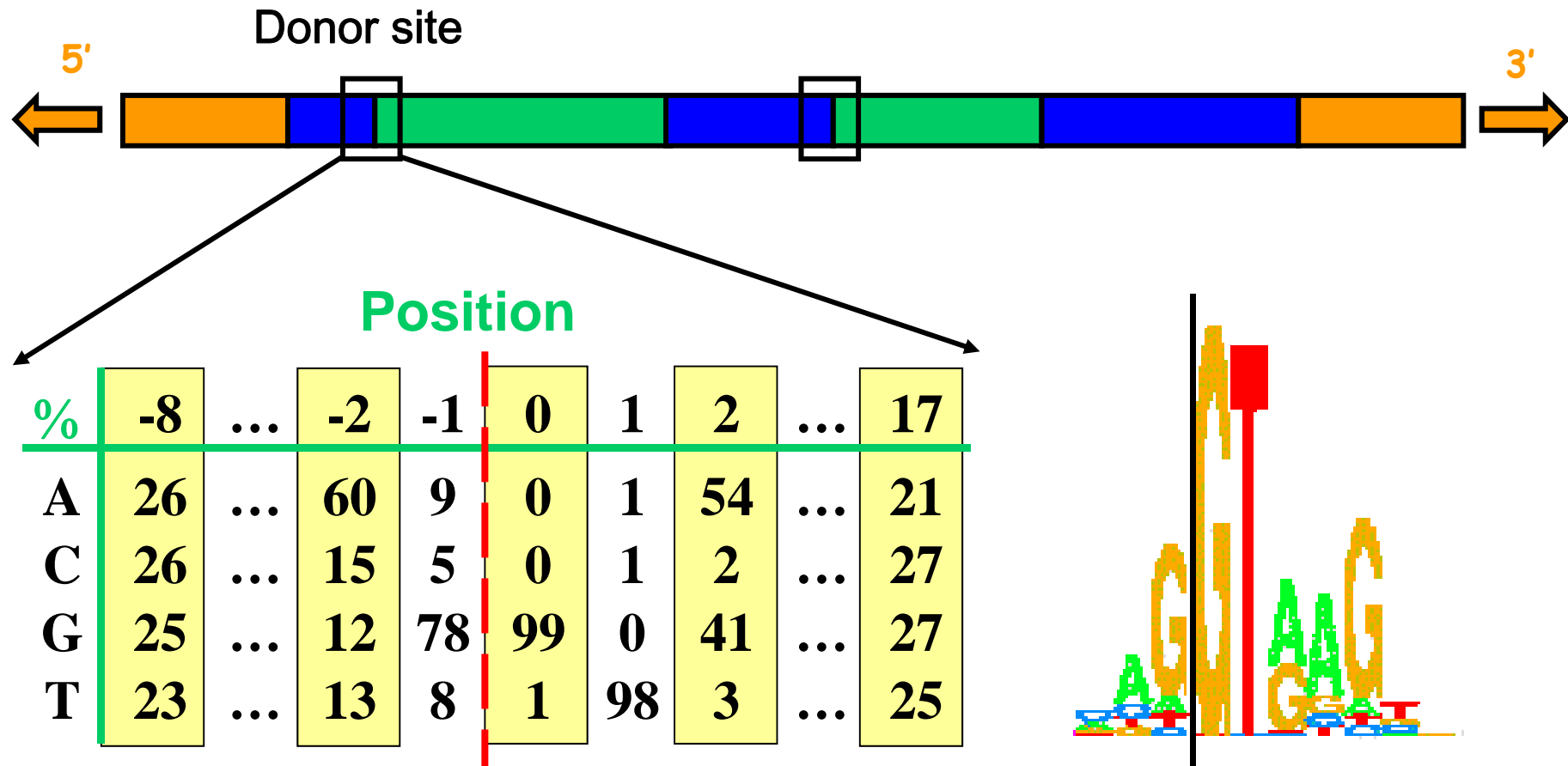
GAGGTAAAC
TCCGTAAGT
CAGGTTGGA
ACAGTCAGT
TAGGTCATT
TAGGTACTG
ATGGTAACT
CAGGTATAC
TGTGTGAGT
AAGGTAAGT

$$\begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}$$



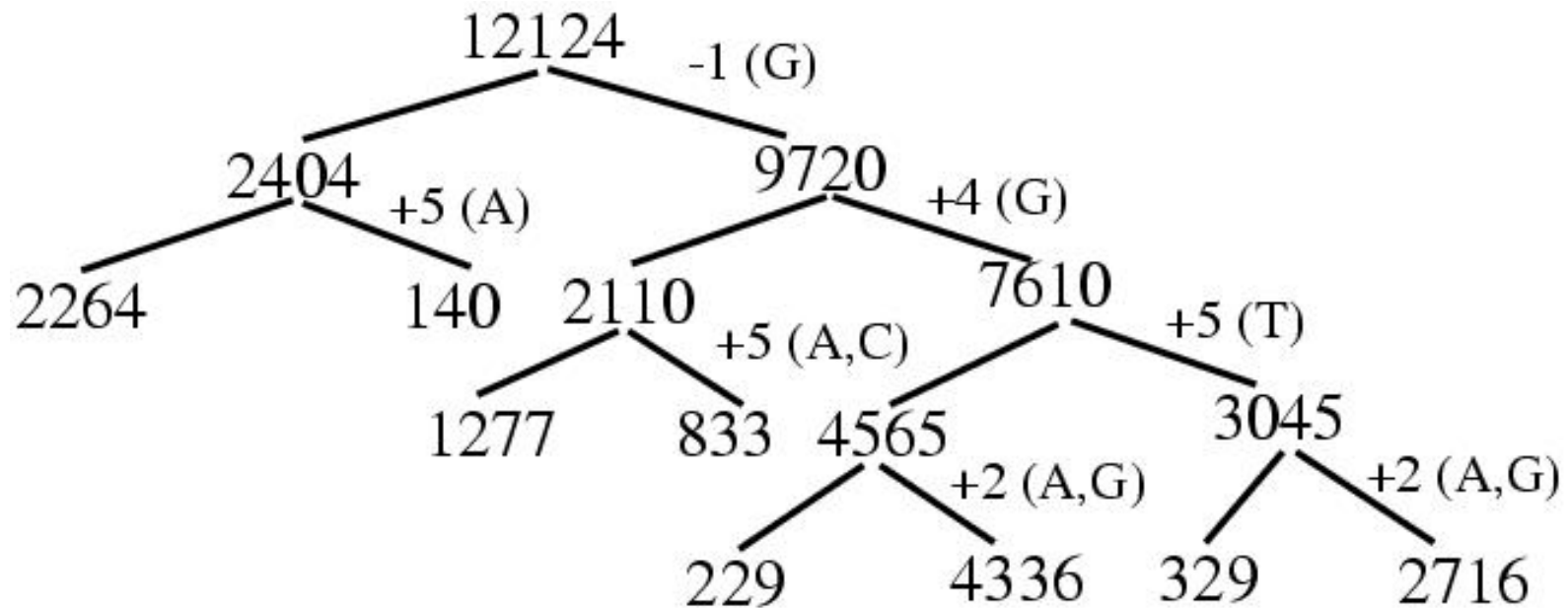
http://en.wikipedia.org/wiki/Position_weight_matrix

Donor Sites: A Simple Model



Donor Sites: An Advance Model

- MDD (maximal dependence decomposition)

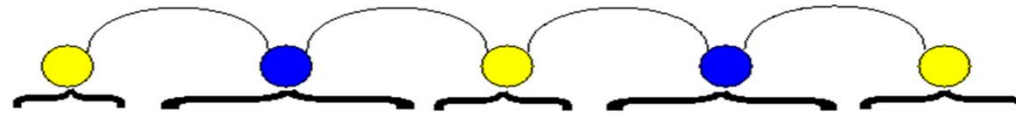


Nodes: Counts

Edges: Split Variables

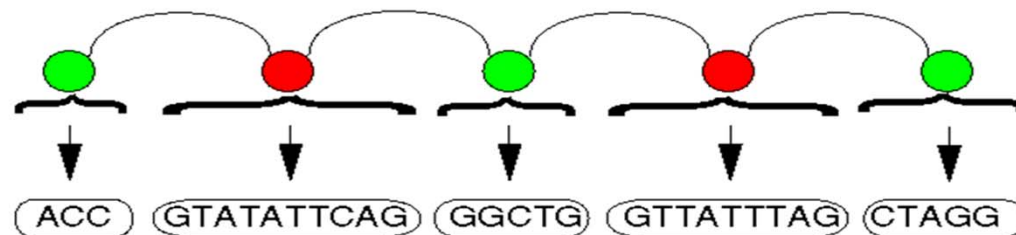
Semi-Markov and Hidden Semi-Markov

Semi-Markov



States have Markov dependence;
each state has an associated length

Hidden Semi-Markov



Hidden states semi-Markov;
observable generated from hidden

Why Semi-Markov Model

- For general Markov model, the duration of state follows a geometric distribution

$$\begin{aligned} P(d = k) \\ &= P(X_n = i, X_{n+1} = i, \dots, X_{n+k-1} = i, X_{n+k} \neq i) \\ &= a_{ii}^k (1 - a_{ii}) \end{aligned}$$

- However, the exon length generally follows a normal distribution

GeneScan

For more detail, see:

- Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268**, 78-94
- Burge, C. B. and Karlin, S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.* **8**, 346-354.

参考文献

- 钱敏平，龚光鲁。《应用随机过程》，北京大学出版社，1998。
- David W. Mount. Bioinformatics, Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, 2002.
- Amy N. Langville and Carl D. Meyer. Deeper Inside PageRank, Internet Mathematics Vol. 1, No. 3: 335-380, 2004.
- L. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proceedings of the IEEE, Vol. 77, No. 2, Feb. 1989
- On-line tutorial: http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html