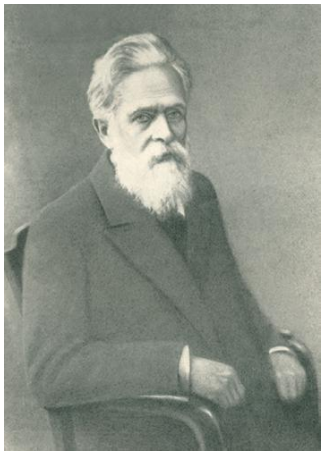
生物统计学：
生物信息中的概率统计模型

2019年秋



有关信息

- 授课教师：宁康
  - Email: ningkang@hust.edu.cn
  - Office: 华中科技大学东十一楼504室
  - Phone: 87793041, 18627968927

- 课程网页
  - http://www.microbioinformatics.org/teach/#
  - QQ群：182996651

课程安排

- 生物背景和课程简介
- 传统生物统计学及其应用
- 生物统计学和生物大数据挖掘
  - Hidden Markov Model (HMM)及其应用
    - Markov Chain
    - HMM理论
    - HMM和基因识别 (Topic I)
    - HMM和序列比对 (Topic II)
  - 进化树的概率模型 (Topic III)
  - Motif finding中的概率模型 (Topic IV)
    - EM algorithm
    - Markov Chain Monte Carlo (MCMC)
  - 基因表达数据分析 (Topic V)
    - 聚类分析-Mixture model
    - Classification-Lasso Based variable selection
  - 基因网络推断 (Topic VI)
    - Bayesian网络
    - Gaussian Graphical Model
  - 基因网络分析 (Topic VII)
    - Network clustering
    - Network Motif
    - Markov random field (MRF)
  - Dimension reduction及其应用 (Topic VIII)
- 面向生物大数据挖掘的深度学习

研究对象:
生物序列,
进化树,
生物网络,
基因表达
...

方法:
生物计算与生物统计

第3章：隐马氏模型(HMM) 及其应用

- Markov Model
- Markov Model的缘起和Page Rank算法
- Hidden Markov Model (HMM)
- HMM的理论基础
- HMM的应用

Part I

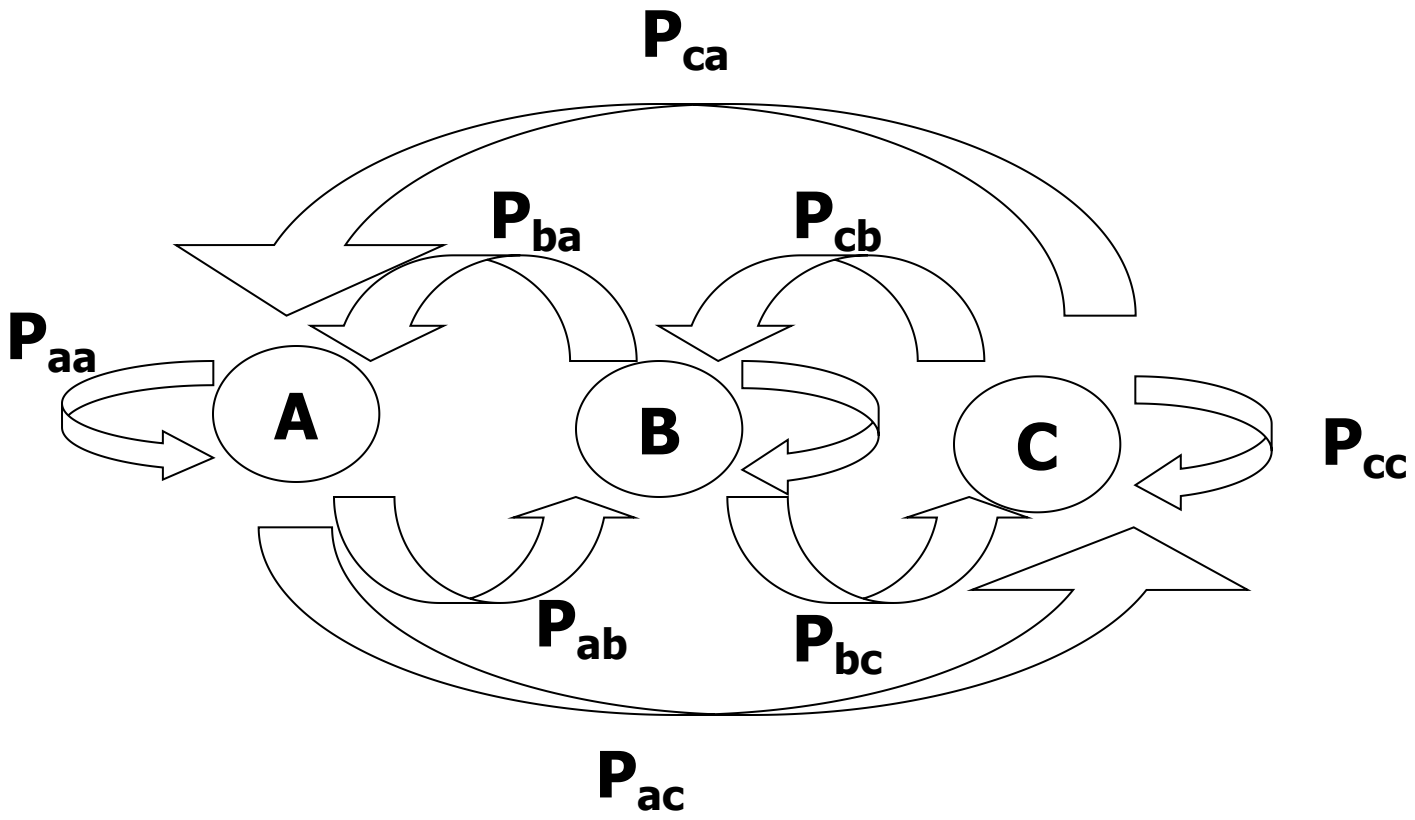
Markov Model

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

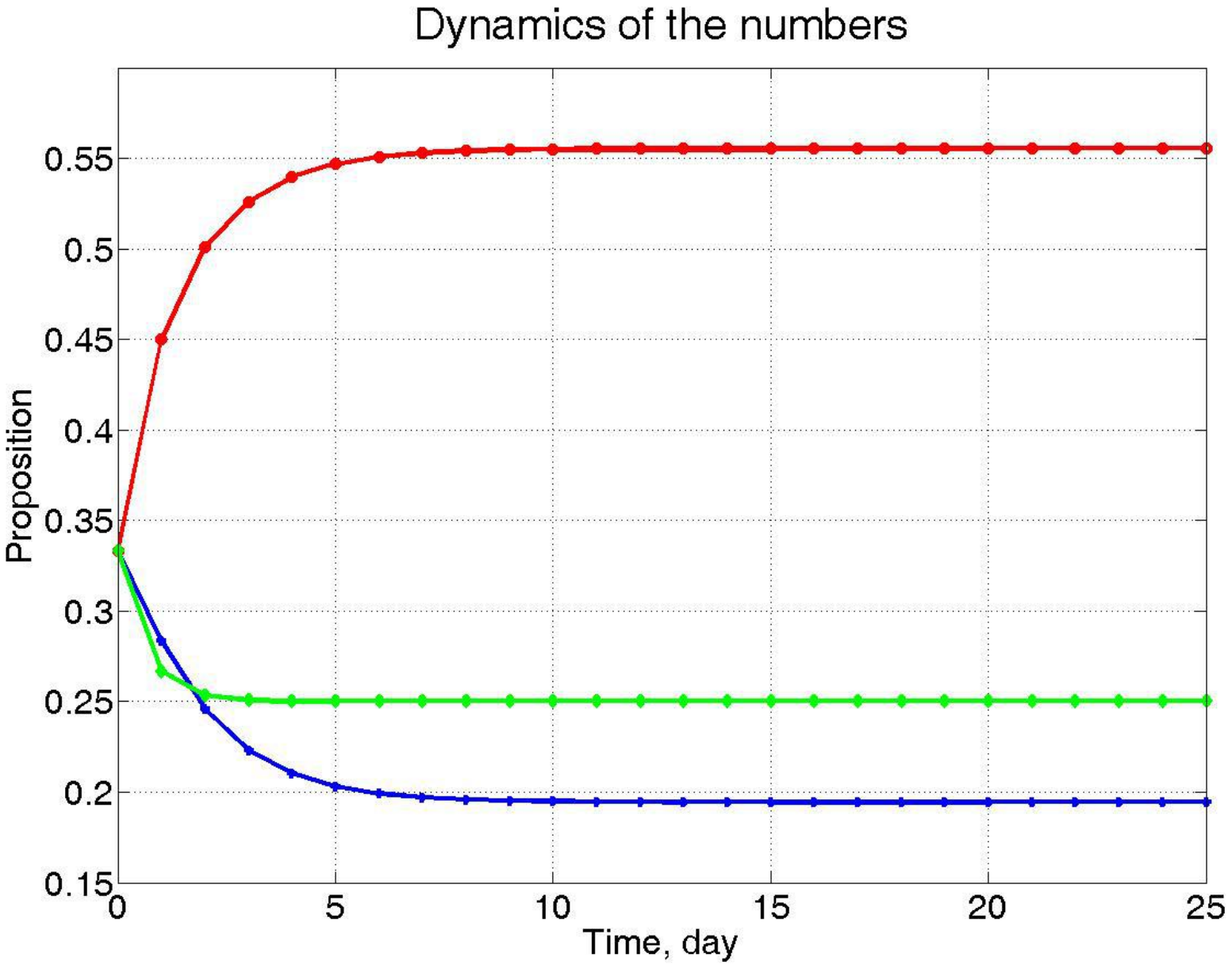
$$(A_n, B_n, C_n) = \pi_0 P^n$$

- 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
Proposition
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0
5
10
15
20
25
Time, day

换起点π=(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, … 的每一个 t 对应一个随机变量$\xi_t(\omega)$，我们把$\xi=\{\xi_0, \xi_1, \cdots, \xi_n, \cdots\}$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, …) 具有公共的取值集合，我们把此集合叫做**状态空间，记为** $\mathscr{S}$。

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

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

- 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：

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

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: $y_n = (x_n, x_{n-1}, \cdots, x_{n-k+1})$，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

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

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 $i_0 = i, i_1, \cdots, i_n = j$, 使得

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

- 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，…， 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

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

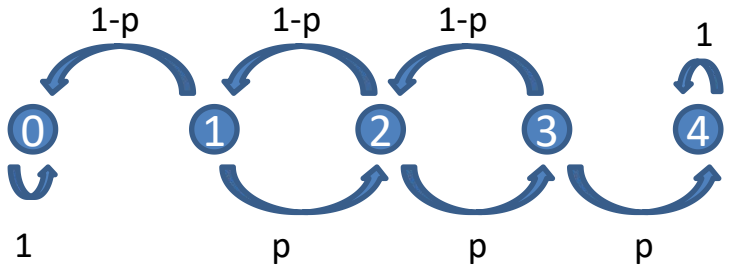
- 这个集合的最大公约数称之为状态i的周期T。

- 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,⋯, 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$\vec{\pi} = (\pi_1, \pi_2, \cdots, \pi_N)$，并使得

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

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,⋯, 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 $\vec{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$，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 $\{\xi_1(\omega), \cdots, \xi_n(\omega)\}$ 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 $\vec{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$,

Part II

Markov Model的缘起和Page Rank算法

Markov Chain

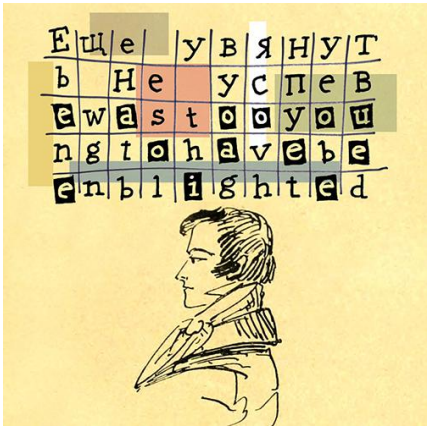
马尔科夫设计马氏链的最初应用，本质上就是机器学习。。。

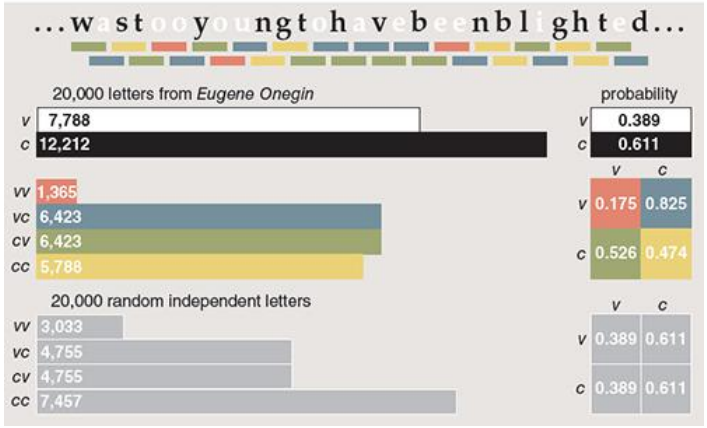
普希金诗作
《叶甫盖尼·奥涅金》

基于统计，确定转移概率参数

马氏链作诗

He was too young to have been blighted
by the cold world's corrupt finesse;
his soul still blossomed out, and lighted
at a friend's word, a girl's caress.
In heart's affairs, a sweet beginner,
he fed on hope's deceptive dinner;
the world's éclat, its thunder-roll,
still captivated his young soul.
He sweetened up with fancy's icing
the uncertainties within his heart;
for him, the objective on life's chart
was still mysterious and enticing—
something to rack his brains about,
suspecting wonders would come out.





First order

Theg sheso pa lyiklg ut. cout Scrpauscricre cobaives wingervet Ners, whe ilened te o wn taulie wom uld atimorerteansouroocono weveiknt hef ia ngry'sif farll t mmat and, tr iscond frnid riliofr th Gureckpeag

Third order

At oness, and no fall makestic to us, infessed Russion-bently our then a man thous always, and toops in he roguestill shoed to dispric! Is Olga's up. Italked fore declaimsel the Juan's conven night toget nothem,

Fifth order

Meanwhile with jealousy bench, and so it was his time. But she trick. Let message we visits at dared here bored my sweet, who sets no inclination, and Homer, so prose, weight, my goods and envy and kin.

Seventh order

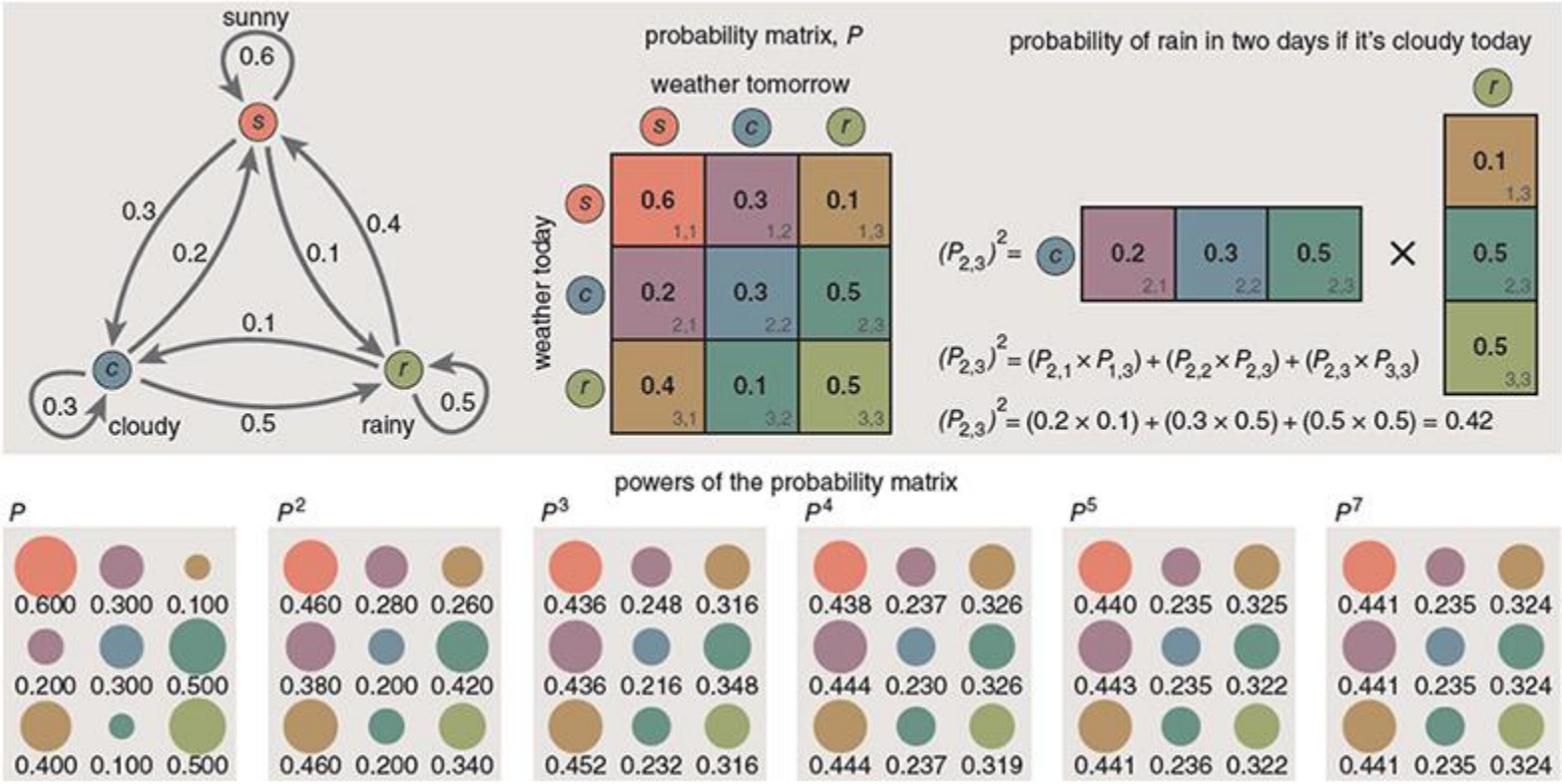
My sorrow her breast, over the dumb torment of her veil, with our poor head is stooping. But now Aurora's crimson finger, your christening glow. Farewell. Evgeny loved one, honoured fate by calmly, not yet seeking?

参考文献：http://html.rhhz.net/kjdb/20180812.htm

Markov Chain

马氏链的广泛应用：Past, Present and Future

基于马氏链的天气预报系统



参考文献：https://www.americanscientist.org/article/first-links-in-the-markov-chain

Markov Chain

- 文学著作的语言统计分析在国内外皆具卓有成效的研究成果。如《罗密欧和朱丽叶》的作者鉴定、《静静的顿河》作者是谁、《红楼梦》前80回和后40回是否皆为曹雪芹所著等。
- 语音识别，图像分析，天气预报等应用。。。

Page Rank

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

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

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.

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

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

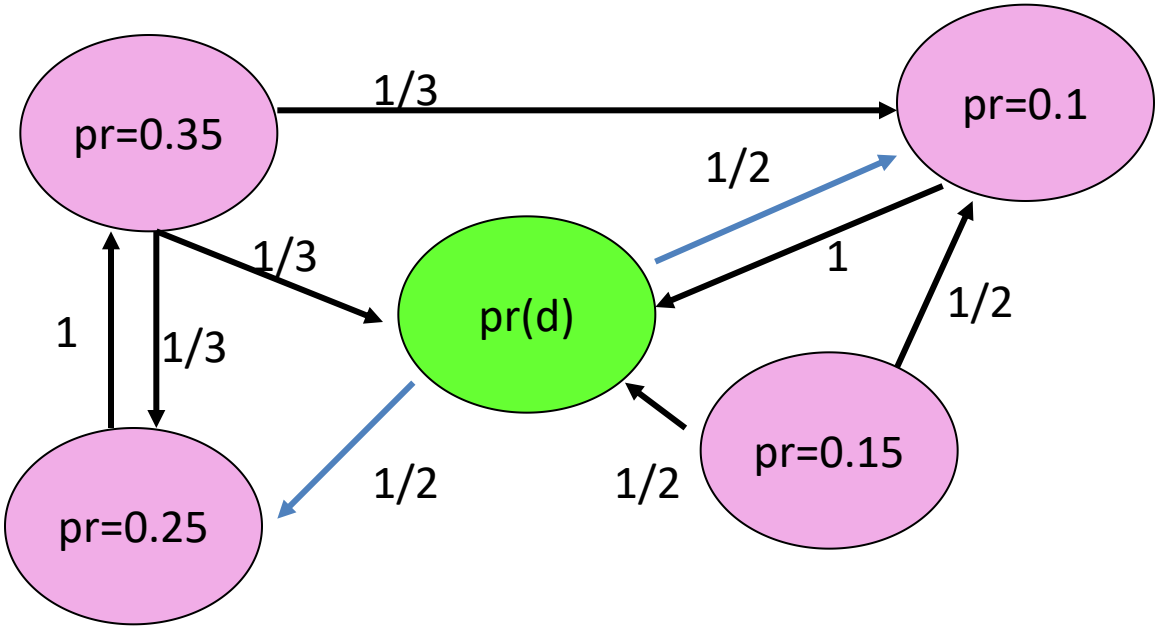
- Once *pr*(*d*) is accepted as a measure of the importance of *d* there is a natural consequence;
- In the calculation of *pr*(*d*), a hyperlink from a page d_1 to *d* should count for more than a hyperlink from page d_2 to *d* if $pr(d_1) > pr(d_2)$
- This motivates:

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

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

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

Simplified Page Rank Calculation



$$pr(d) = 0.35 * 1/3 + 0.1 * 1 + 0.15 * 1/2 = 0.292$$

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

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,

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

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

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.

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

Markov Chain Interpretation

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

- 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.

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

Damping Factor

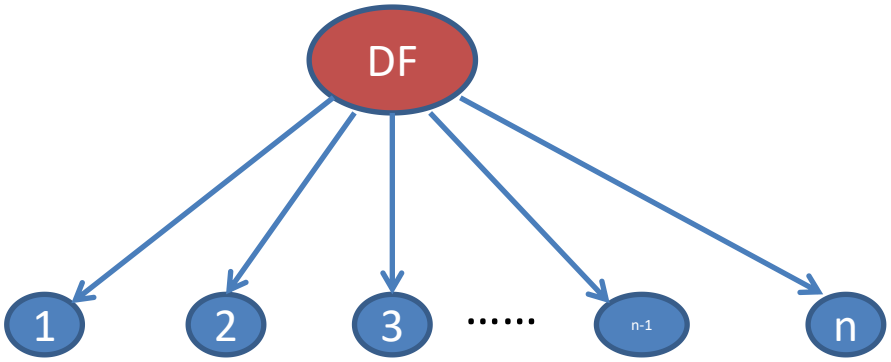
- It represents the change that user stopping clicking links and get bored with the current page and then request another random page.
- If the damping factor is 85% then there is assumed to be about a 15% chance that a typical users won't follow any links on the page and instead navigate to a new random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

Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

Page Rank

- Taking into account the damping factor

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

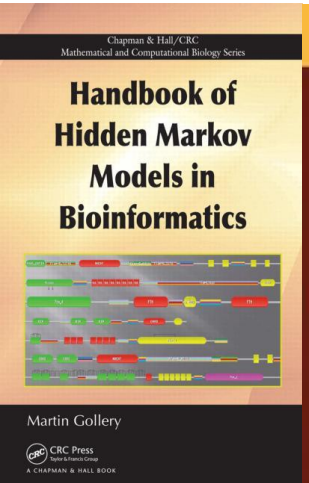


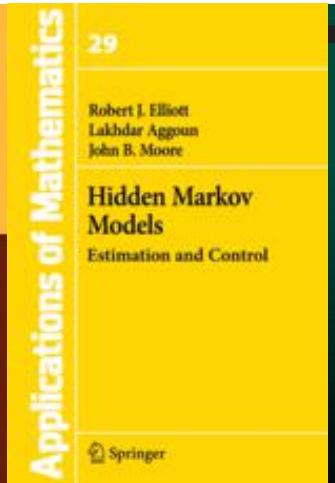
Modified from http://www.eee.bham.ac.uk/russellm/ee3j2/Slides%202009/OHP10%20Page%20Rank%202009.pdf

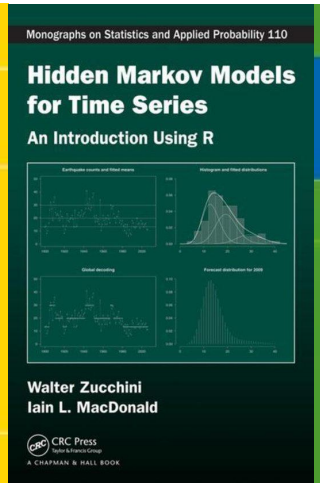
Part III

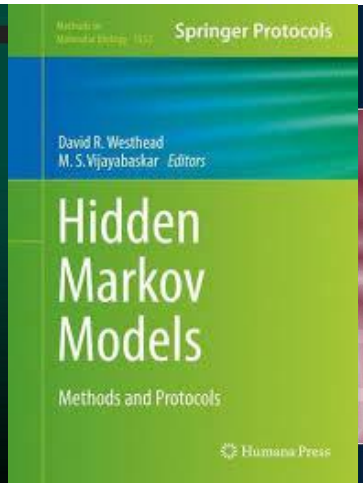
Hidden Markov Model

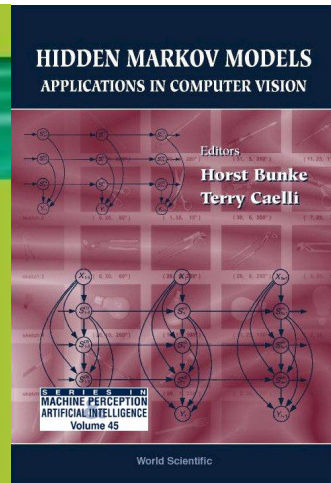
Hidden Markov Models - HMM



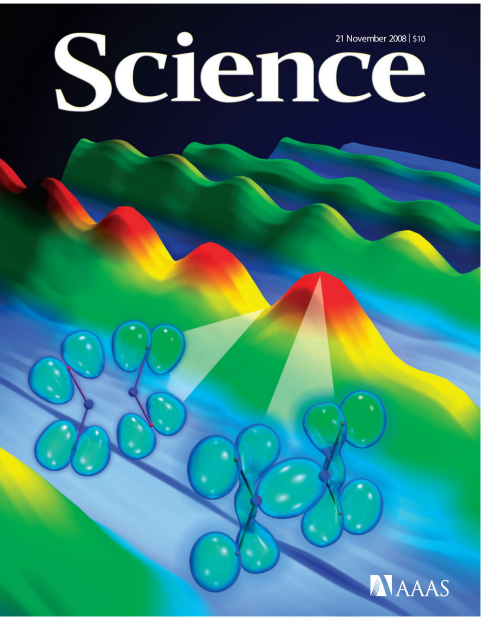


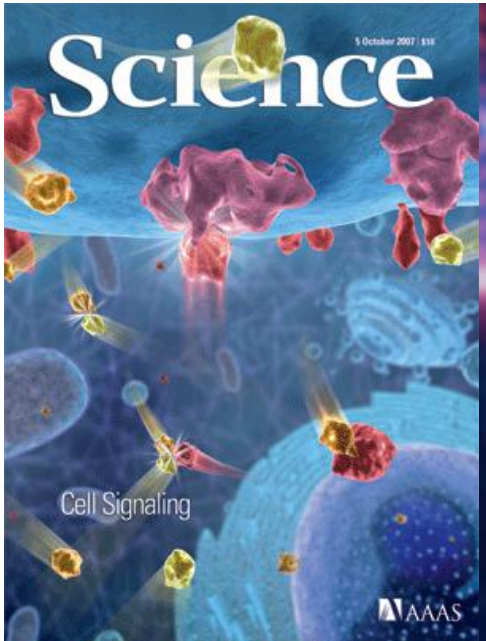


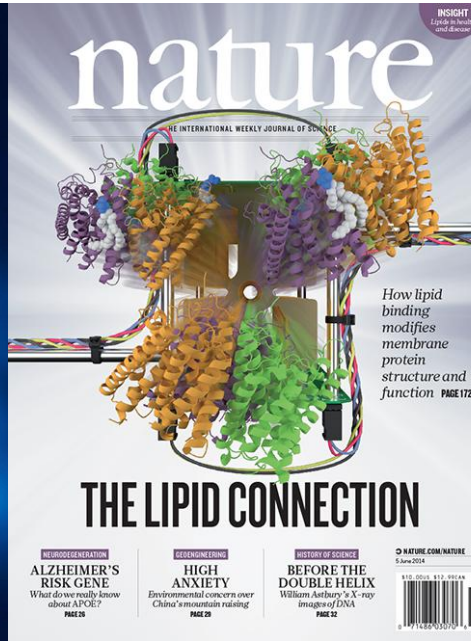




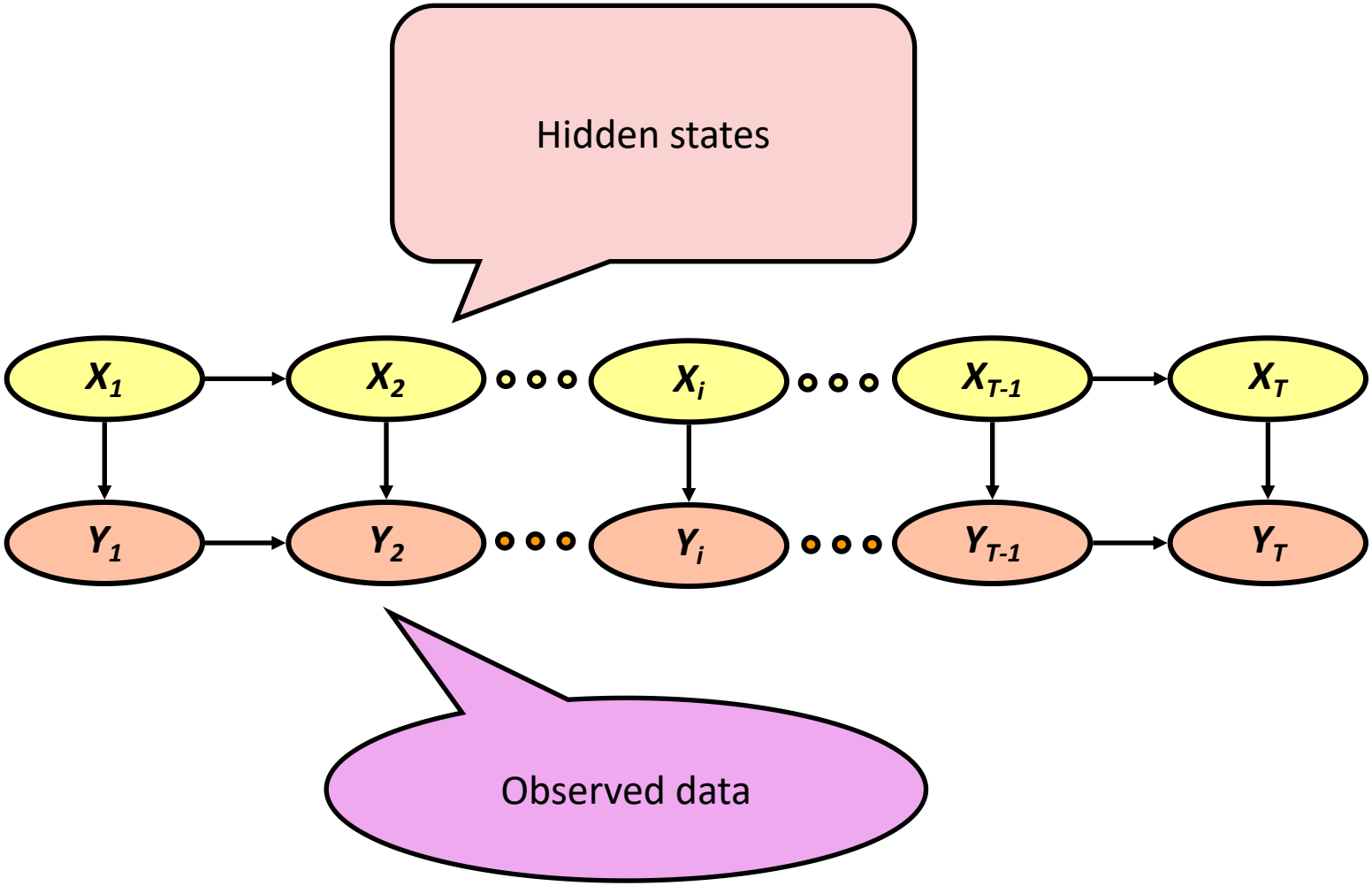
Hidden Markov Models - HMM



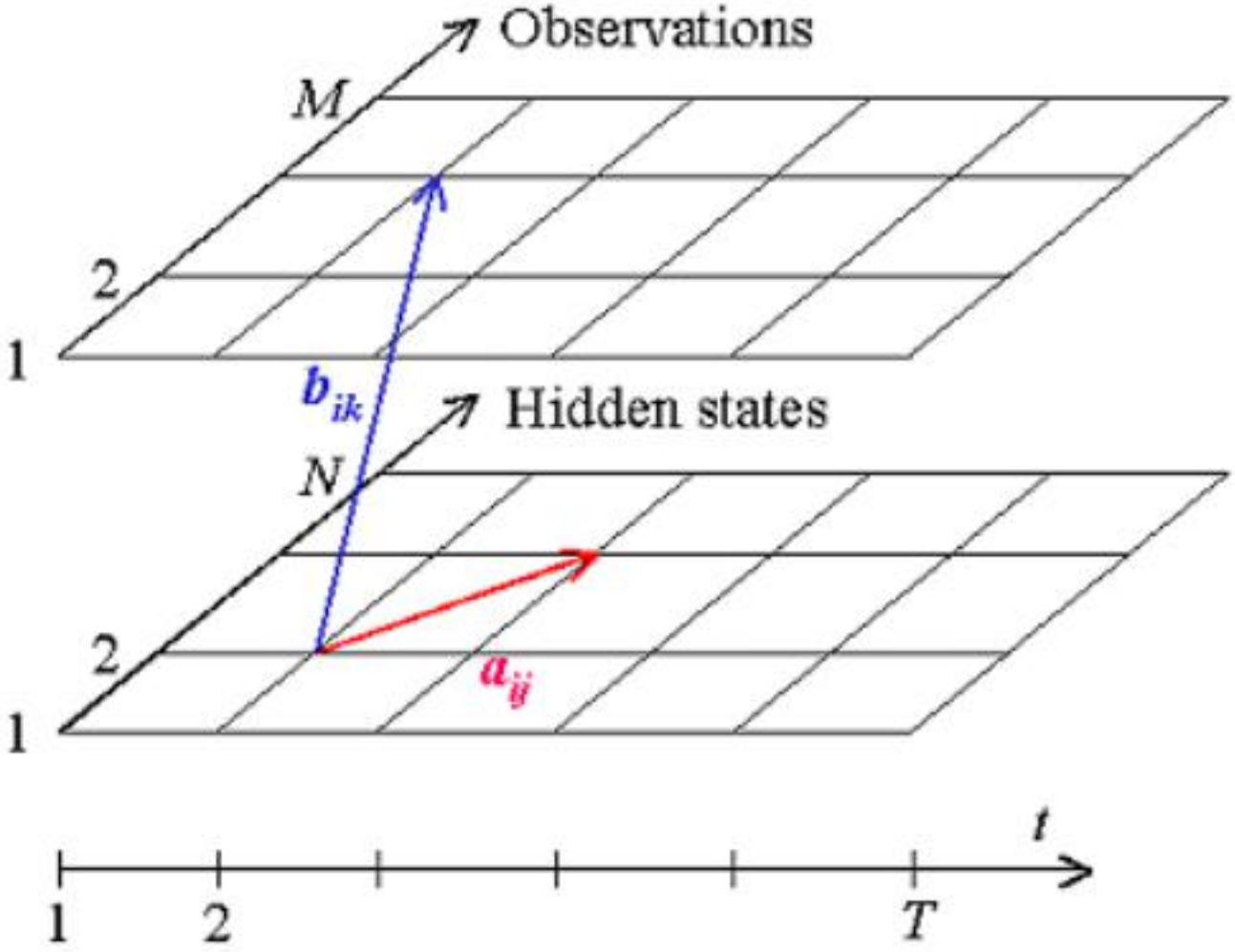




Hidden Markov Models - HMM

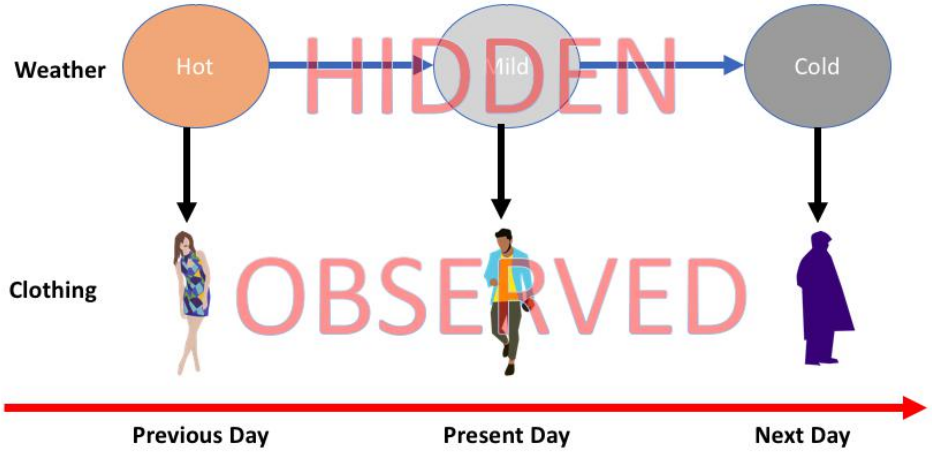


Hidden Markov Models - HMM



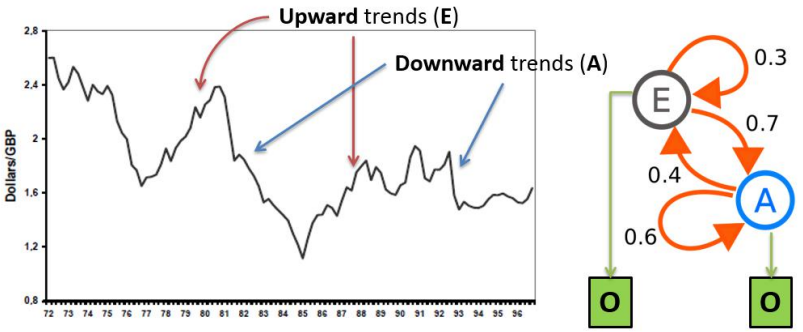
Hidden Markov Models - HMM

Whether forcast

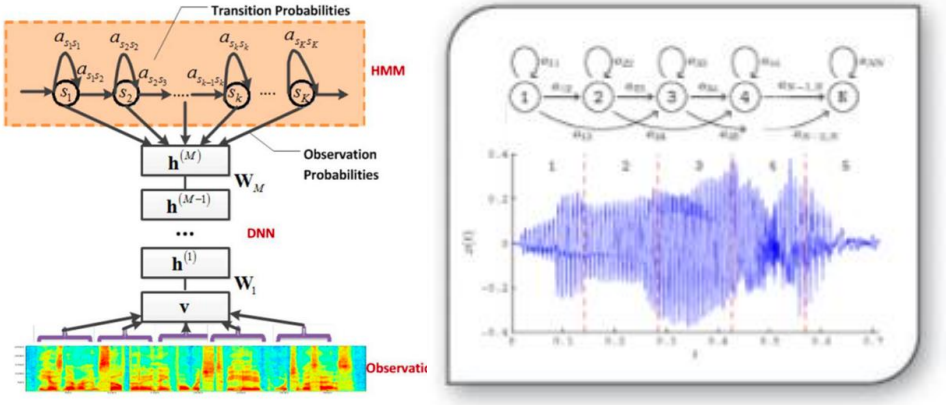


Finance prediction

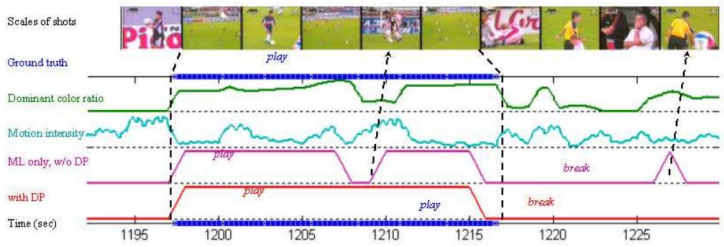
Modeling the **hidden "regimes"** of financial markets – switches between periods of **high volatility** & **low volatility**, **bearish** & **bullish**, etc.



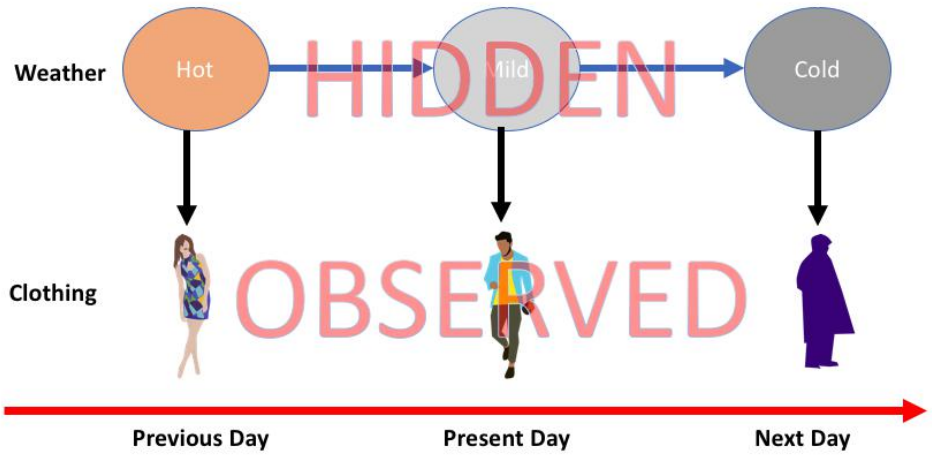
Acoustic analysis

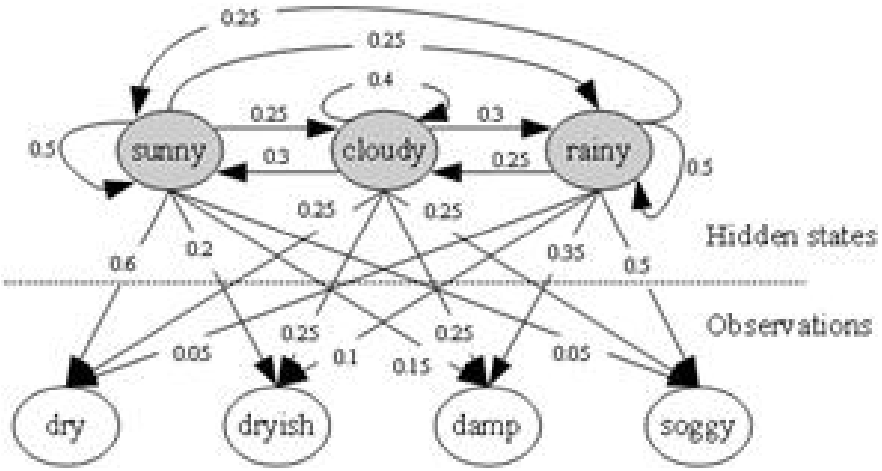


Video analysis



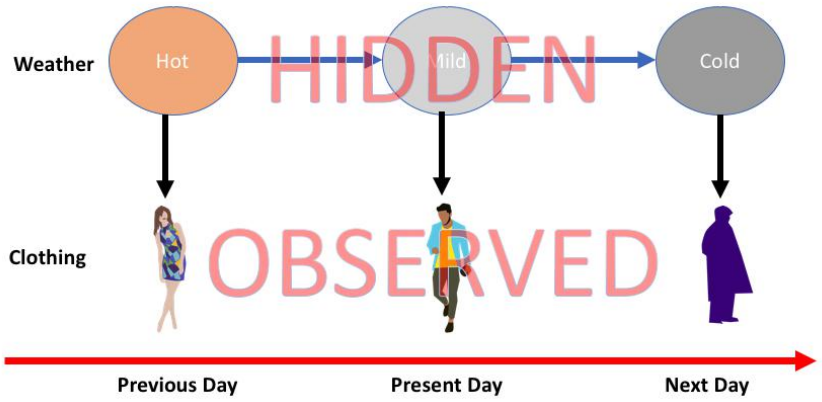
Hidden Markov Models - HMM

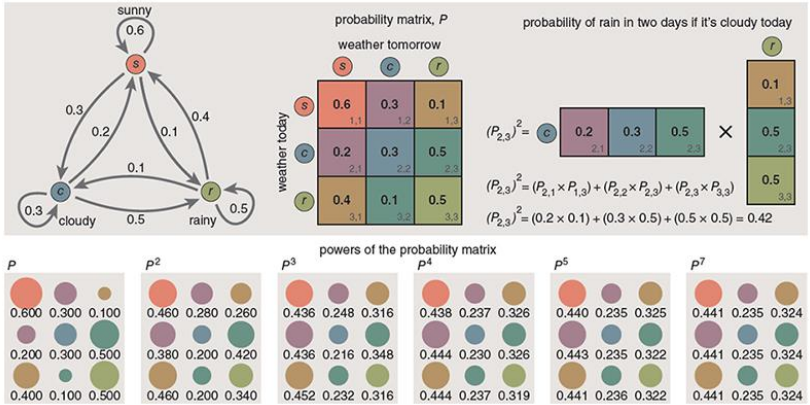




- HMM has more sets of parameters
- But only this?

HMM vs. Markov Chain





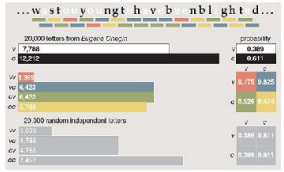
- The questions are different
- HMM do not need to know the states
- The types of known parameters are also different

HMM vs. Markov Chain

The questions are different

普希金诗作
《叶甫盖尼·奥涅金》

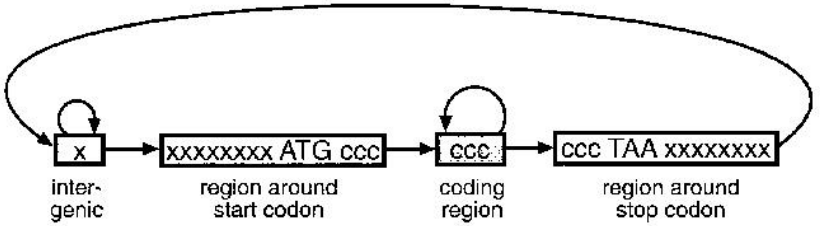
基于统计，确定转移概率参数



马氏链作诗

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

A simple HMM for unspliced genes:

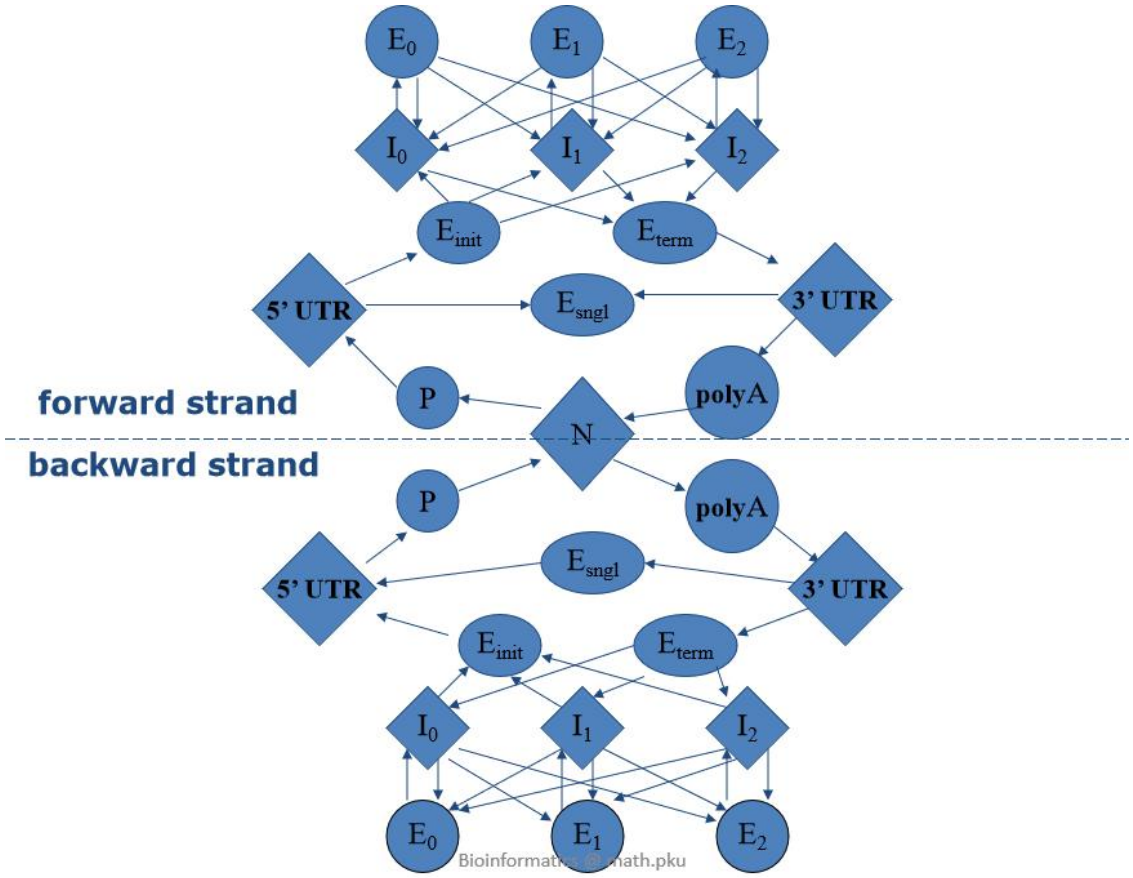


AAAGC ATG CAT TTA ACG AGA GCA CAA GGG CTC TAA TGCCG

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

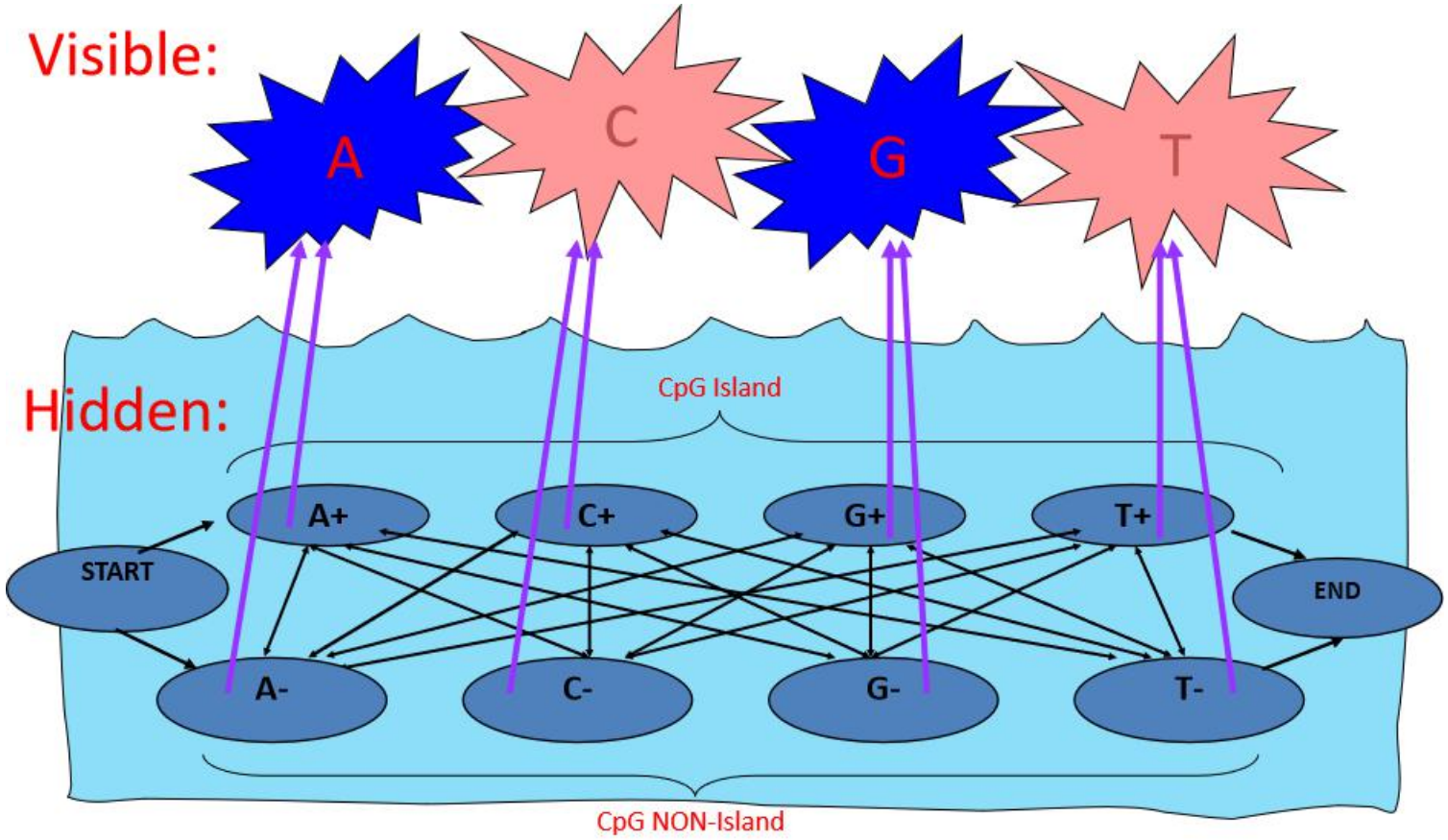
HMM vs. Markov Chain

HMM do not need to know the states



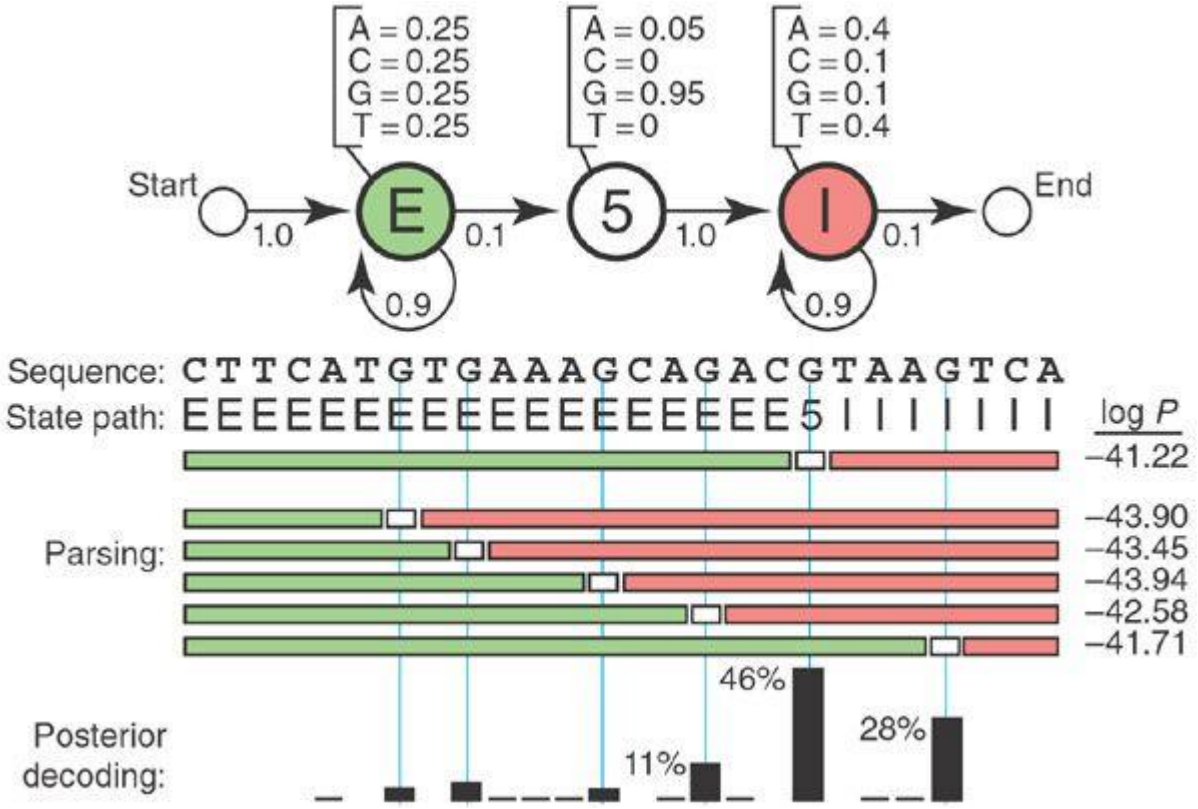
HMM vs. Markov Chain

The types of known parameters are also different



Hidden Markov Models - HMM

A toy HMM: 5′ splice site recognition



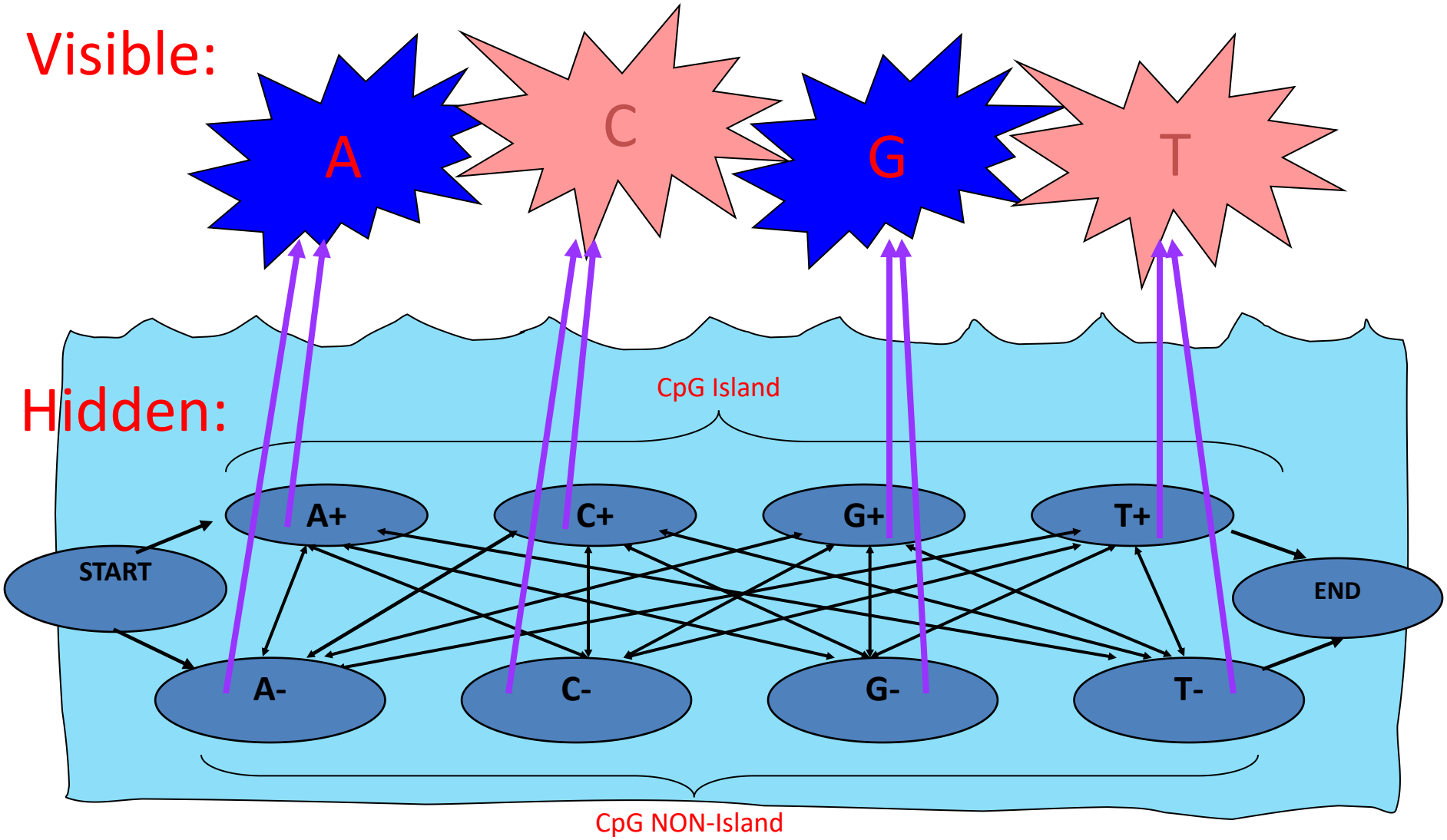
Assumption:

exons have a uniform base composition on average (25% each base), introns are A/T rich (say, 40% each for A/T, 10% each for C/G), and the 5′SS consensus nucleotide is almost always a G (say, 95% G and 5% A).

So, what's hidden?

Reference: https://www.nature.com/articles/nbt1004-1315

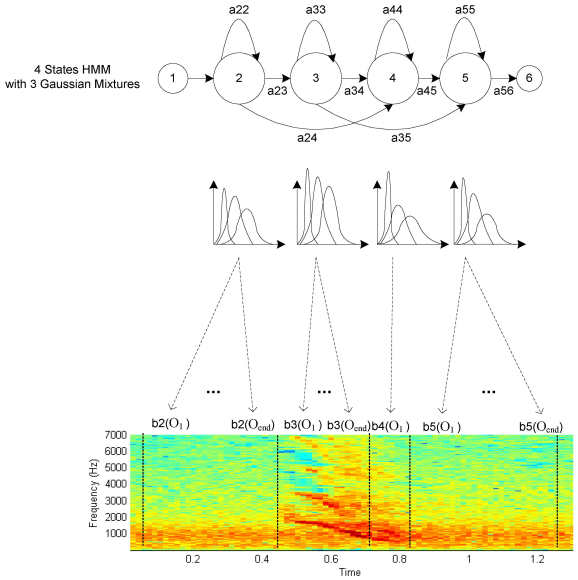
Hidden Markov Models - HMM



Hidden Markov Models - HMM

More exsamples in biology

Gene expression



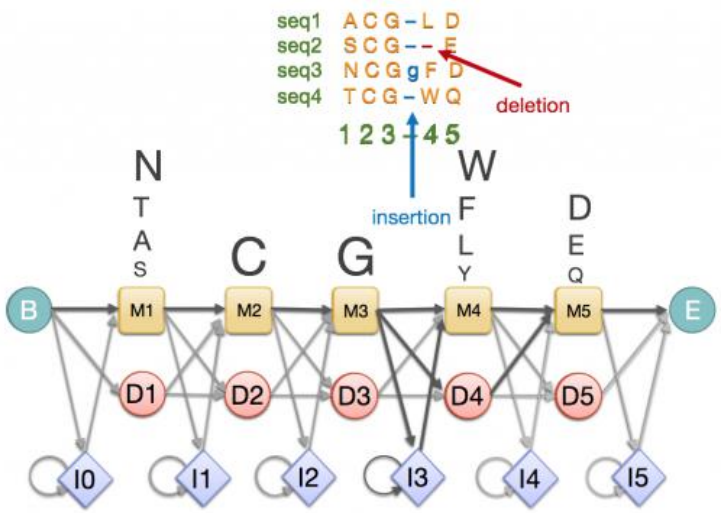
Protein sequence alignment

Start with a multiple sequence alignment

Insertions / deletions can be modelled

Occupancy and amino acid frequency at each position in the alignment are encoded

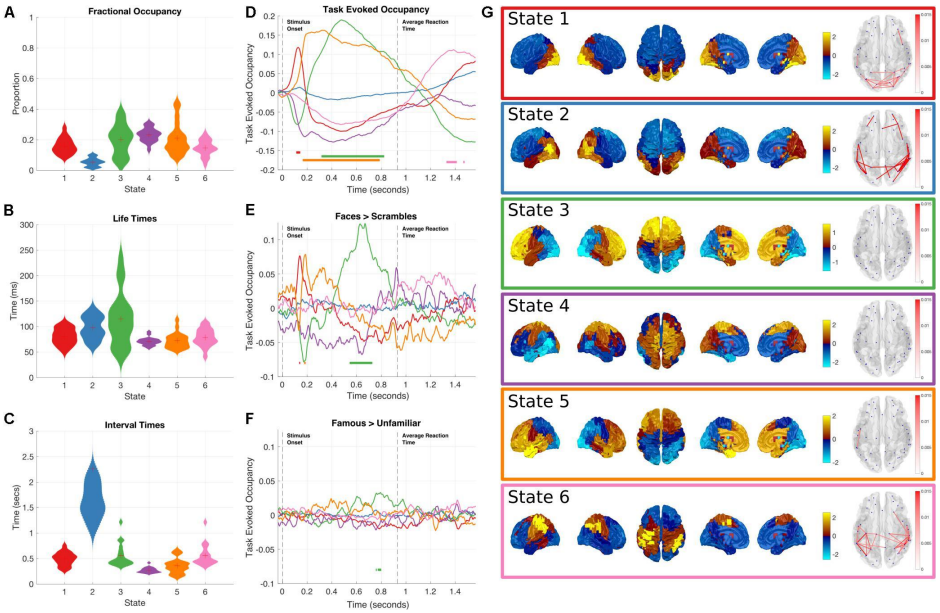
Profile created



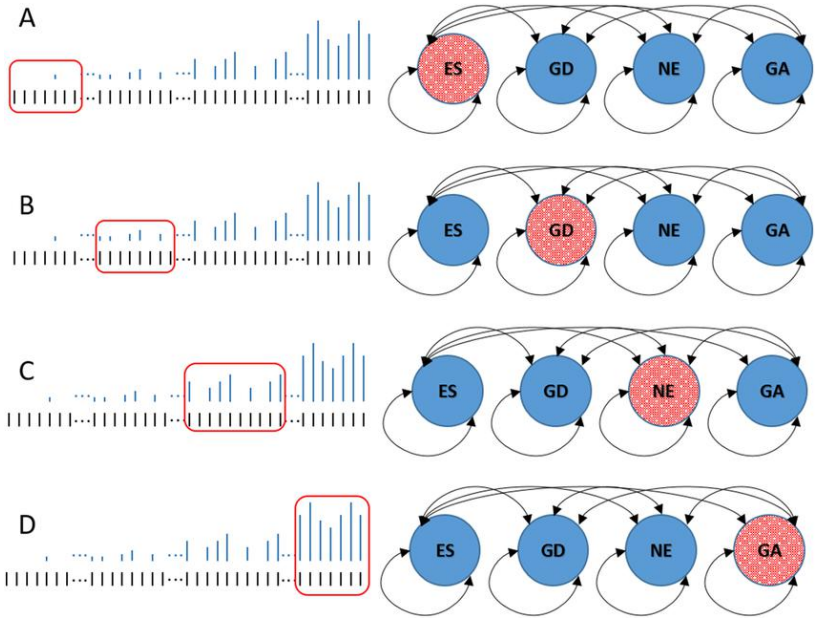
Hidden Markov Models - HMM

More exsamples in biology

Brain sceince
(image, electro-signal, etc.)

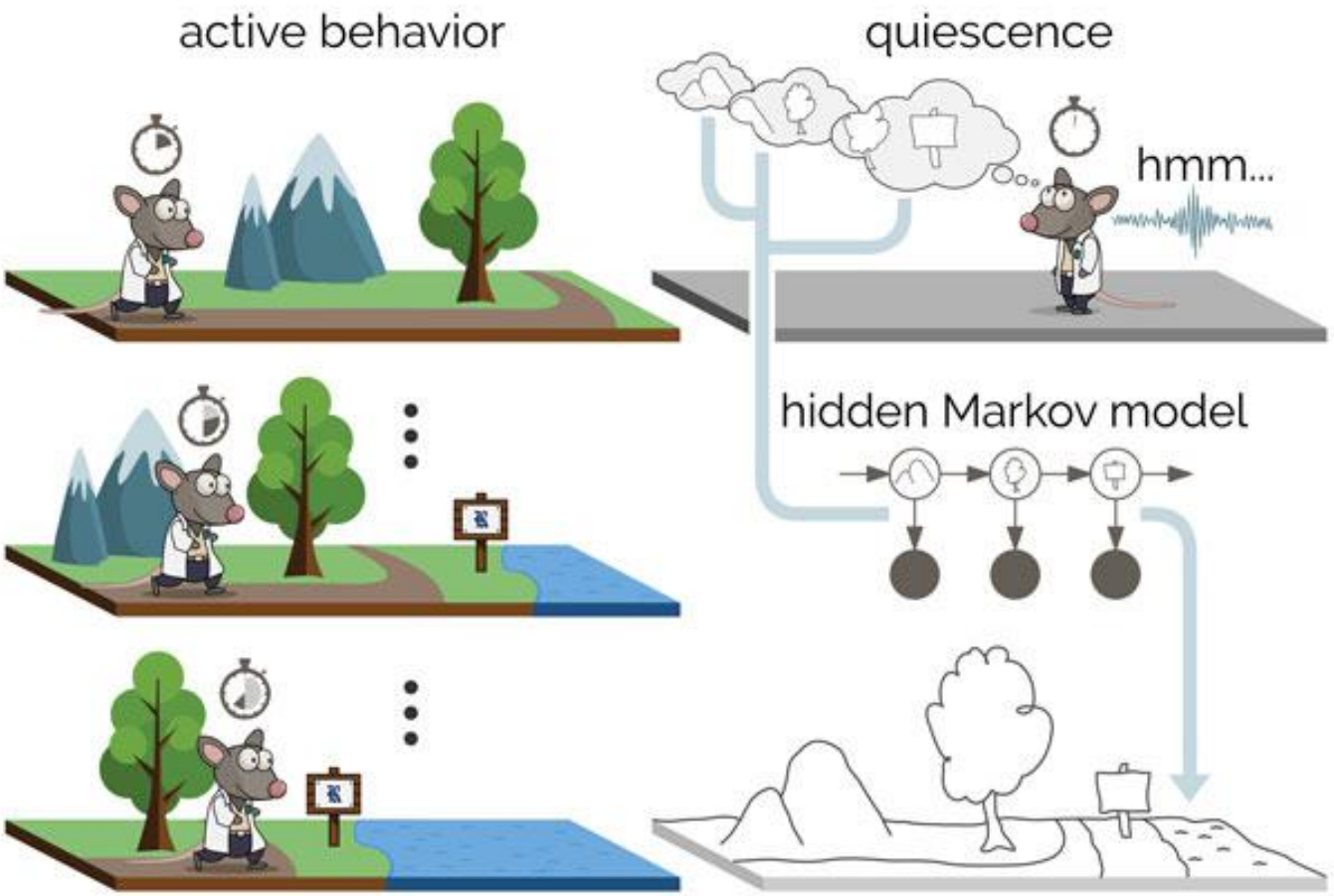


Public health



Hidden Markov Models - HMM

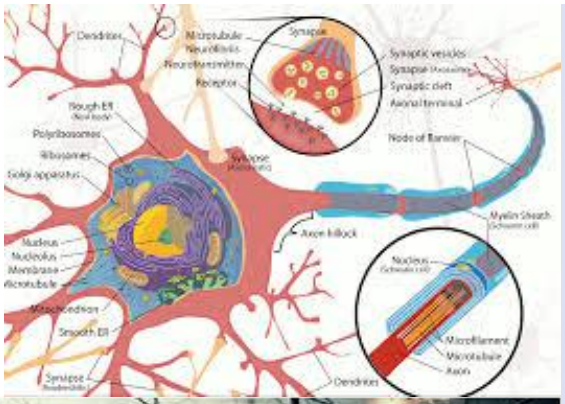
We are actually HMM animal...

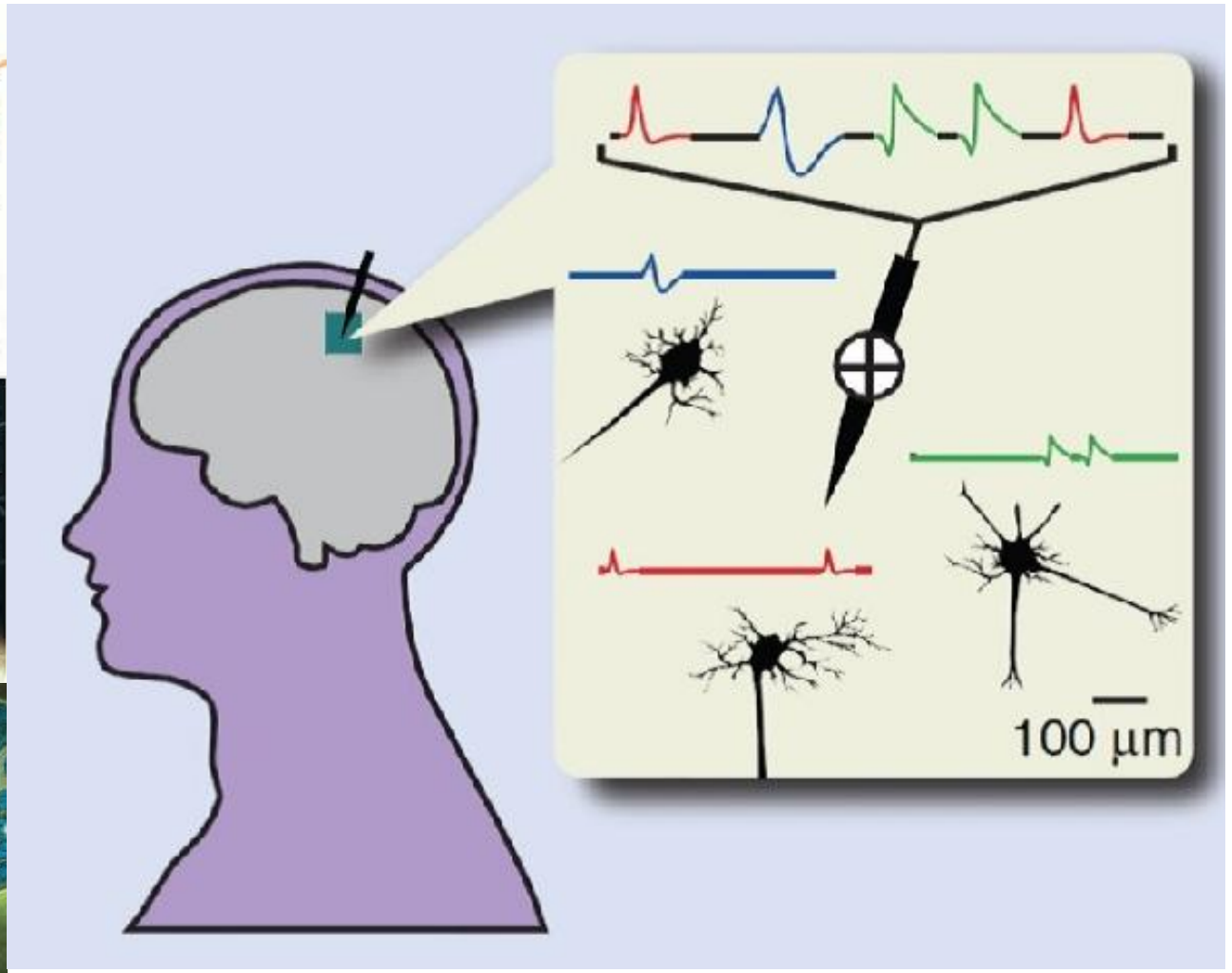


Reference: https://engineering.rice.edu/news/neurons-ripple-while-brains-rest-lock-memories

Hidden Markov Models - HMM

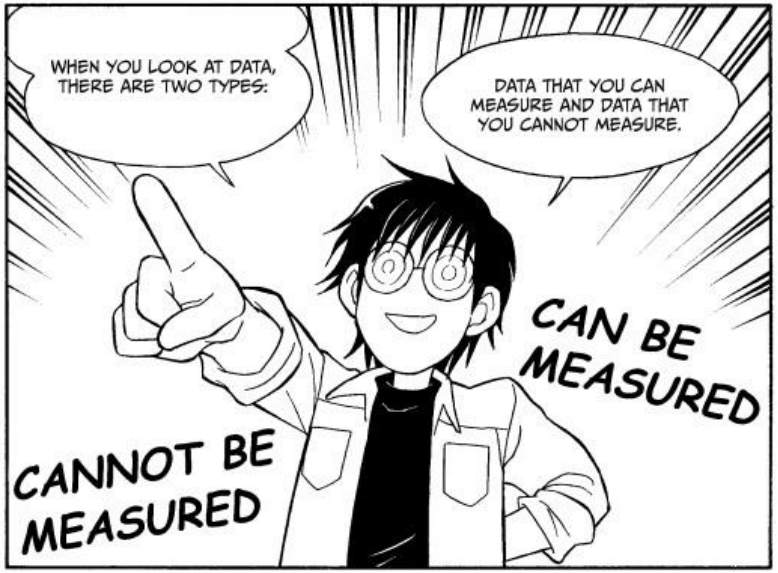
We are actually HMM animal...





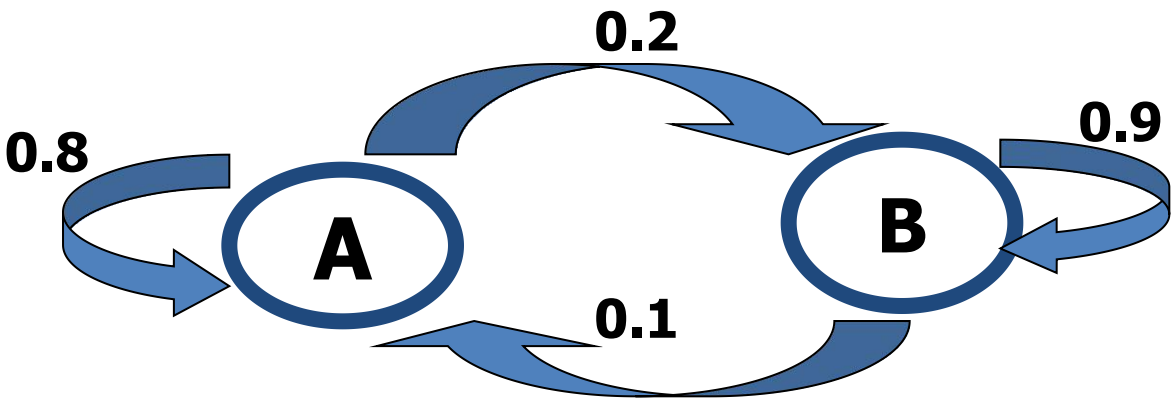
So, lets start to learn HMM!

"Data don't make any sense,
we will have to resort to statistics."



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$\boldsymbol{\lambda = \{\pi, A, B\}}$
  - 初始分布$\boldsymbol{\pi}=(\pi_i)$, $\pi_i=P\{X_1=i\}$
  - 转移矩阵$\boldsymbol{A}= (a_{ij})$, $a_{ij} = P(X_{n+1}=j \mid X_n= i)$
  - 给定某个时间的隐状态的条件下，观测的分布矩阵$\boldsymbol{B}=(b_{il})$, $b_{il} = P(Y_n=l \mid X_n= i)$。

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 | λ), 归一化就得到给定样本来自每个模型的概率 P (λ | 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

$$Pr(Y=y|\lambda)=\sum_{X=x} Pr(Y=y|X=x,\lambda)Pr(X=x|\lambda)$$

$$=\sum_{x=(x_1,\cdots,x_T)} \pi(x_1)b_{x_1}(y_1)a_{x_1x_2}b_{x_2}(y_2)\cdots a_{x_{T-1}x_T}b_{x_T}(y_T)$$

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

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

- 迭代

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

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

- 结果

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

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

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

- 迭代

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

$$1 \leq i \leq N, \quad t = T-1, \cdots, 1;$$

- 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 $Y = (y_1, y_2, \cdots, y_T)$, 如何给出隐状态序列 $X^0 = (x_1^0, x_2^0, \cdots, x_T^0)$.
- 单点最优

$$\gamma_t(i) = Pr(X_t = i|Y)$$
$$X_t' = \underset{i}{\text{Argmax}}\, \gamma_t(i)$$

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

$$Pr(x_1', x_2', \cdots, x_T'|y_1, \cdots, y_T)$$
$$\geq Pr(x_1, x_2, \cdots, x_T|y_1, \cdots, y_T)$$

解码问题(II)

- 由Bayesian公式有

$$Pr(x'_1, x'_2, \cdots, x'_T | y_1, \cdots, y_T)$$
$$= \frac{Pr(x_1, x_2, \cdots, x_T, y_1, \cdots, y_T)}{Pr(y_1, \cdots, y_T)}$$

- 又由于序列 **Y** 给定, 问题等价于找最优的X°使联合概率 $Pr(x_1, \cdots, x_T; y_1, \cdots, y_T)$ 最大。

最优单点确定

$$\gamma_t(i) = Pr(X_t = i|y_1, y_2, \cdots, y_T, \lambda)$$

$$= \frac{Pr(X_t = i, y_1, \cdots, y_T|\lambda)}{Pr(y_1, \cdots, y_T|\lambda)}$$

$$= \frac{Pr(X_t = i, y_1, y_2, \cdots, y_T|\lambda)}{\sum_i Pr(X_t = i, y_1, y_2, \cdots, y_T|\lambda)}$$

$$= \frac{Pr(y_{t+1}, \cdots, y_T|x_t = i, y_1, \cdots, y_t, \lambda) Pr(x_t = i, y_1, \cdots, y_t|\lambda)}{\sum_i Pr(X_t = i, y_1, y_2, \cdots, y_T|\lambda)}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_i \alpha_t(i)\beta_t(i)}$$

Viterbi算法

Exciting things are happening here.

Viterbi算法(I)

- 算法的思想动态规划的递推算法。
- 递推变量为

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

- 我们有递推公式

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

- 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

$$\delta_1(i) = \pi_i b_i(y_1),$$
$$\psi_1(i) = 0, \quad i = 1, 2, \cdots, N.$$

- 迭代

$$\delta_t(j) = \left(\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij} \right) b_j(y_t)$$
$$\psi_t(j) = \operatorname*{Argmax}_{1 \le i \le N} \left(\delta_{t-1}(i) a_{ij} \right)$$
$$t = 2, \cdots, T; \ j = 1, \cdots, N.$$

Viterbi算法(III)

- 终止

$$p^\star = \max_{1\le i\le N} \delta_T(i)$$

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

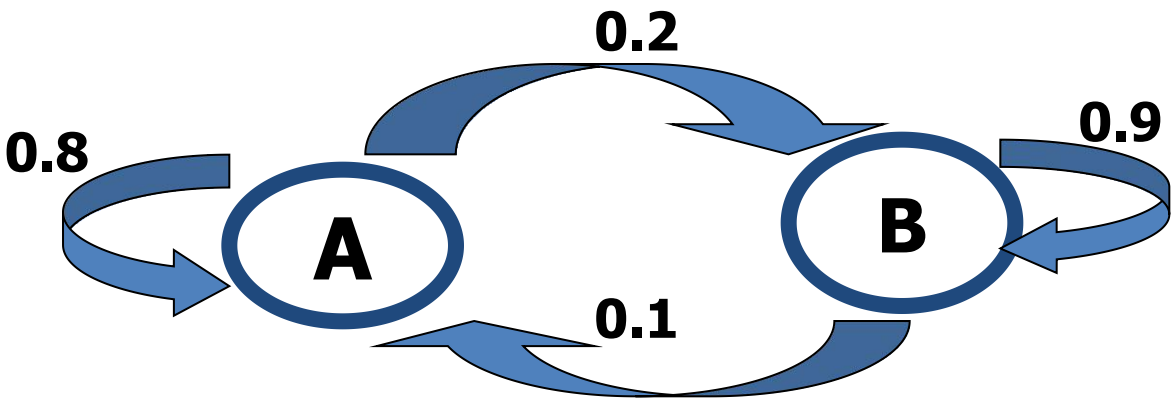
- 后推

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

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

Viterbi算法实例(I)

- 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



Viterbi算法实例(I)

- 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？

Viterbi算法实例(I)

- 转移概率以及初概率(π, A)

	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) (B)

	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	2.371×10^{-4}	A	1.067×10^{-4}	A
t=5	5	3.161×10^{-4}	A	2.881×10^{-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

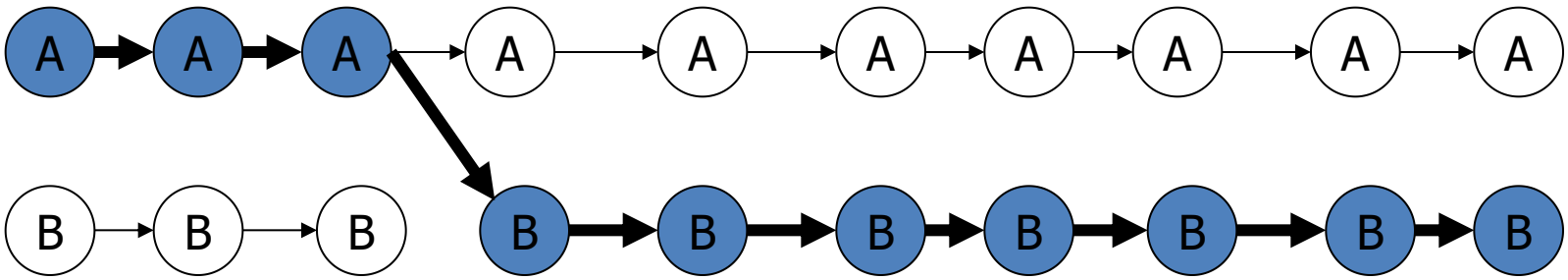
Viterbi算法实例(II)

	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	2.371×10^{-4}	A	1.067×10^{-4}	A
t=5	5	3.161×10^{-4}	A	2.881×10^{-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

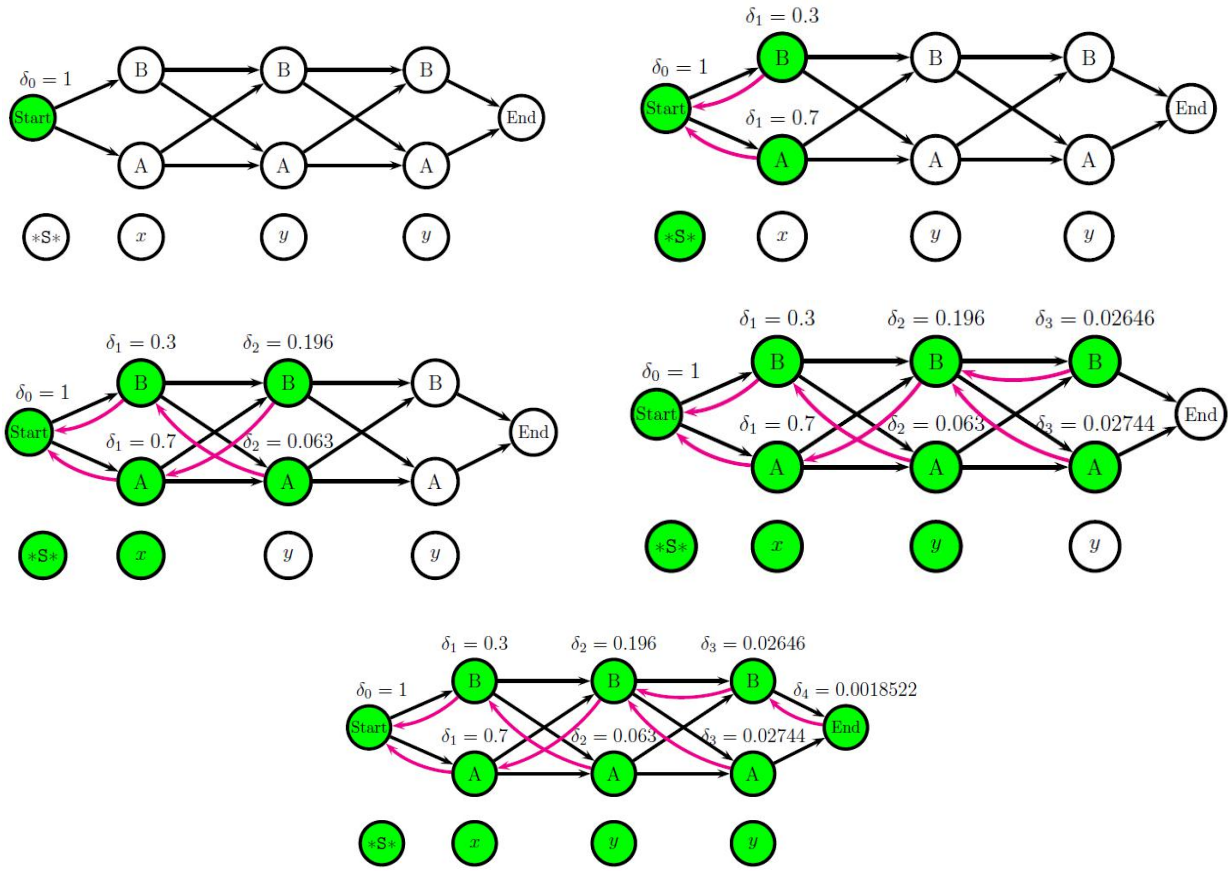


解码出来的状态序列为：

A A A B B B B B B B

Viterbi算法

也可以每一步立即实施解码：



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)

$$\widehat{\lambda} = \underset{\lambda}{\text{Argmax}}\, Pr(y_1, \cdots, y_T|\lambda)$$

- 状态链已知时

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

- 状态链未知时

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

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

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

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

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

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

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

- 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

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

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（最大期望算法）

- 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（最大期望算法）

- **EM算法例子：**

小时候，老妈给一大袋糖果给你，叫你和你姐姐等分，然后你懒得去点糖果的个数，所以你也就不知道每个人到底该分多少个。咱们一般怎么做呢？

（1）先把一袋糖果目测的分为两袋
（2）然后把两袋糖果拿在左右手，看哪个重（E-step）
（3）如果右手重，那很明显右手这代糖果多了，然后你再在右手这袋糖果中抓一把放到左手这袋（M-step）
（4）然后再感受下哪个重（E-step）
（5）然后再从重的那袋抓一小把放进轻的那一袋（M-step）
（6~n）继续下去，直到你感觉两袋糖果差不多相等了为止（收敛）

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).

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

- M-step (取极大).

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

MLE算法（最大似然估计算法）

- 最大似然估计：已知某个随机样本满足某种概率分布，但是其中具体的参数不清楚，参数估计就是通过若干次试验，观察其结果，利用结果推出参数的大概值。

- 最大似然估计是建立在这样的思想上：已知某个参数能使这个样本出现的概率最大，我们就把这个参数作为估计的真实值。

- 时间复杂度极大

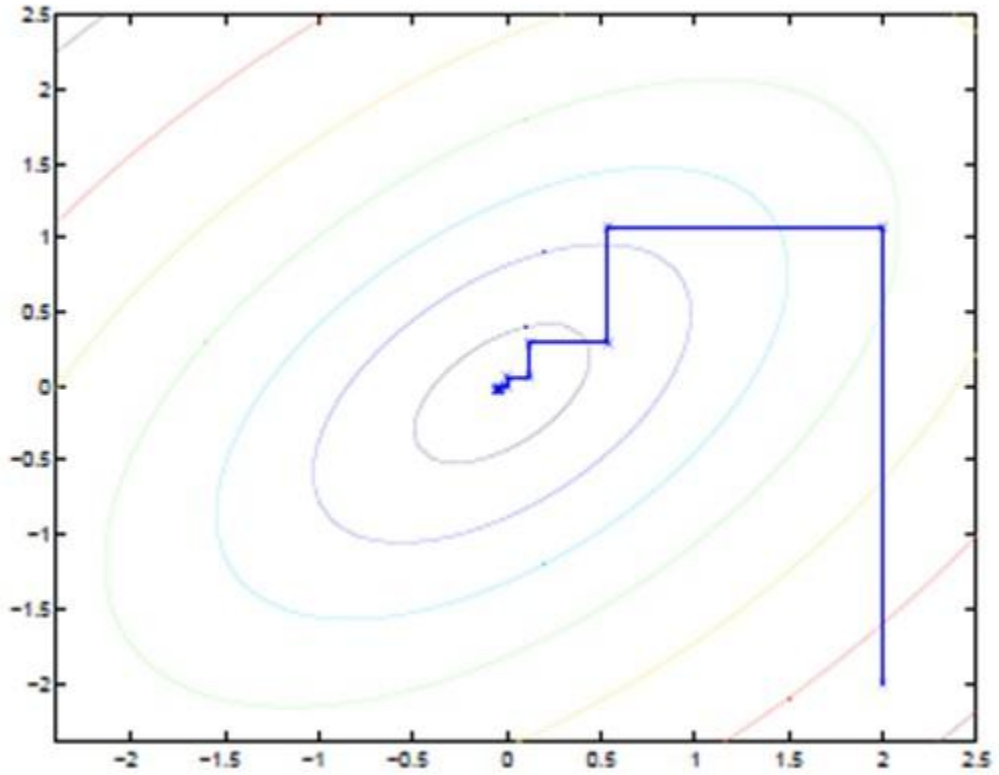
EM算法（最大期望算法）

- 最大期望算法：假设我们想估计知道A和B两个参数，在开始状态下二者都是未知的，但如果知道了A的信息就可以得到B的信息，反过来知道了B也就得到了A。

- 首先赋予A某种初值，以此得到B的估计值，然后从B的当前值出发，重新估计A的取值，这个过程一直持续到收敛为止。

- 多项式时间复杂度 

EM算法（最大期望算法）



这犹如在x-y坐标系中找一个曲线的极值，然而曲线函数不能直接求导，因此什么梯度下降方法就不适用了。但固定一个变量后，另外一个可以通过求导得到，因此可以使用坐标上升法，一次固定一个变量，对另外的求极值，最后逐步逼近极值。对应到EM上，E步：固定θ，优化Q；M步：固定Q，优化θ；交替将极值推向最大。

MLE算法 vs. EM算法（Examples）

- **MLE算法**

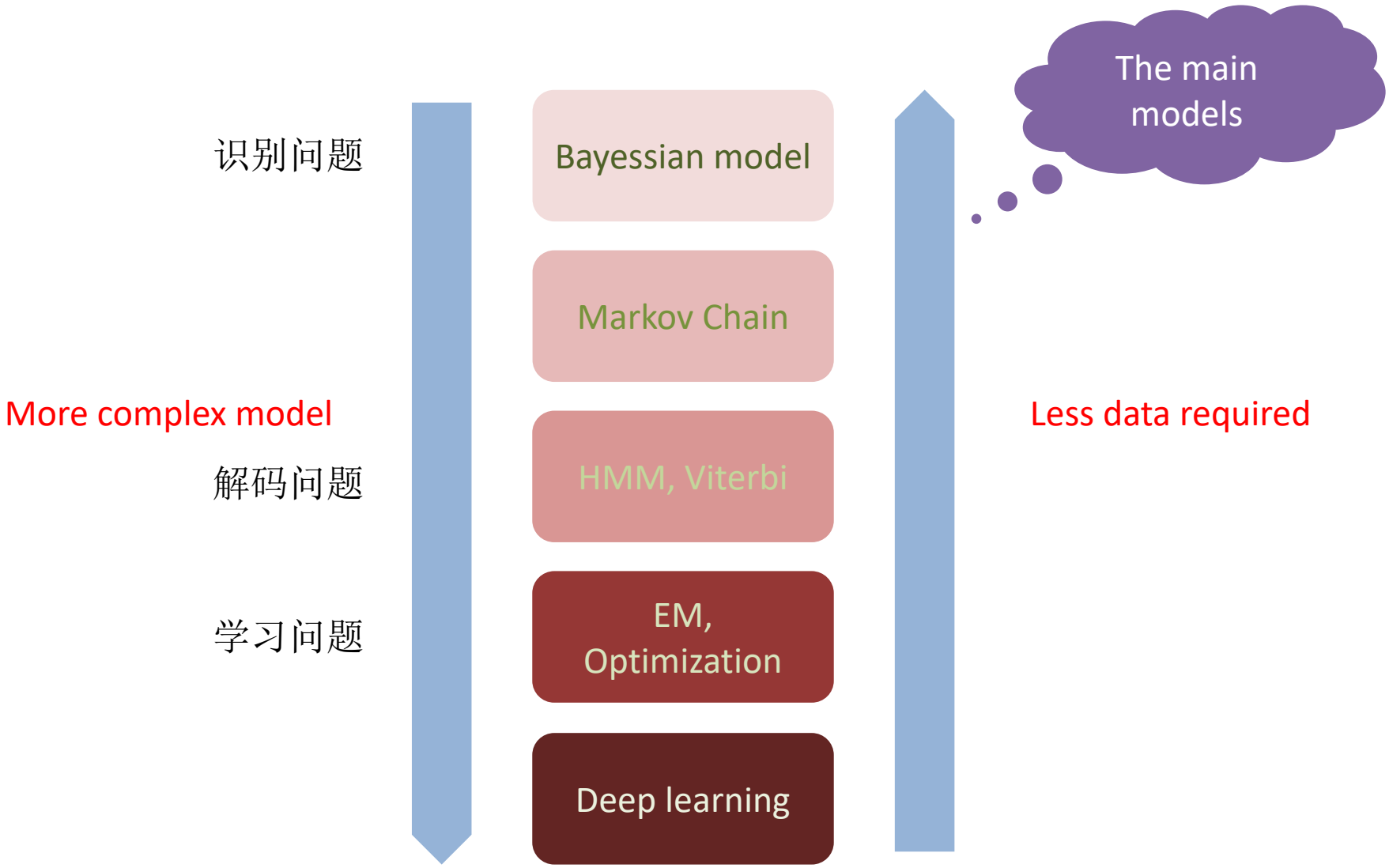
某位同学与一位猎人一起外出打猎，一只野兔从前方窜过。只听一声枪响，野兔应声到下，如果要你推测，这一发命中的子弹是谁打的？你就会想，只发一枪便打中，由于猎人命中的概率一般大于这位同学命中的概率，看来这一枪是猎人射中的。

- **EM算法**

小时候，老妈给一大袋糖果给你，叫你和你姐姐等分，然后你懒得去点糖果的个数，所以你也就不知道每个人到底该分多少个。咱们一般怎么做呢？先把一袋糖果目测的分为两袋，然后把两袋糖果拿在左右手，看哪个重，如果右手重，那很明显右手这代糖果多了，然后你再在右手这袋糖果中抓一把放到左手这袋，然后再感受下哪个重，然后再从重的那袋抓一小把放进轻的那一袋，继续下去，直到你感觉两袋糖果差不多相等了为止。

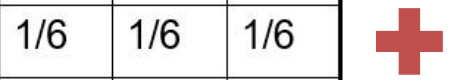
Reference: https://blog.csdn.net/zouxy09/article/details/8537620

Statistical modeling



Statistical modeling

	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

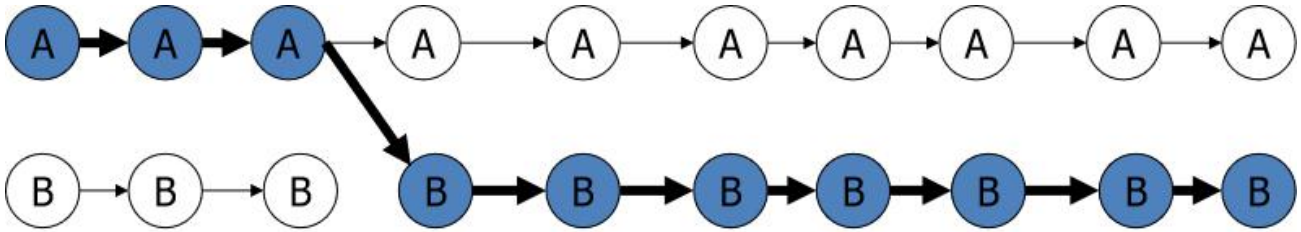


	A	B
A	0.8	0.2
B	0.1	0.9
初概率	0.6	0.4

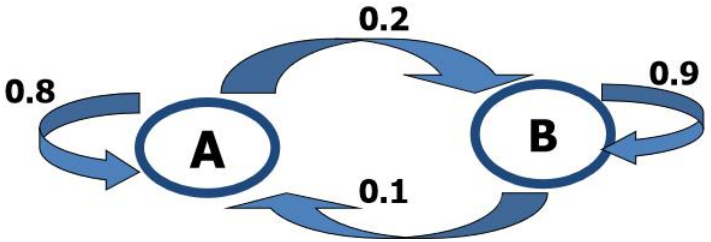
识别问题

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

解码问题



学习问题



?

我们学会了HMM算法！


Well done!
Good job!
You did it.
Good for you!
That is great.
That is good.
Sensational!
What a good try.
That is clever.
Tremendous.
Outstanding!
Fantastic.
Nice going.
I am impressed.
I knew you could do it.
Keep up the good work.
CONGRATULATIONS
English Study Page

Part V

隐马氏模型应用

应用1：基因序列CpG岛识别

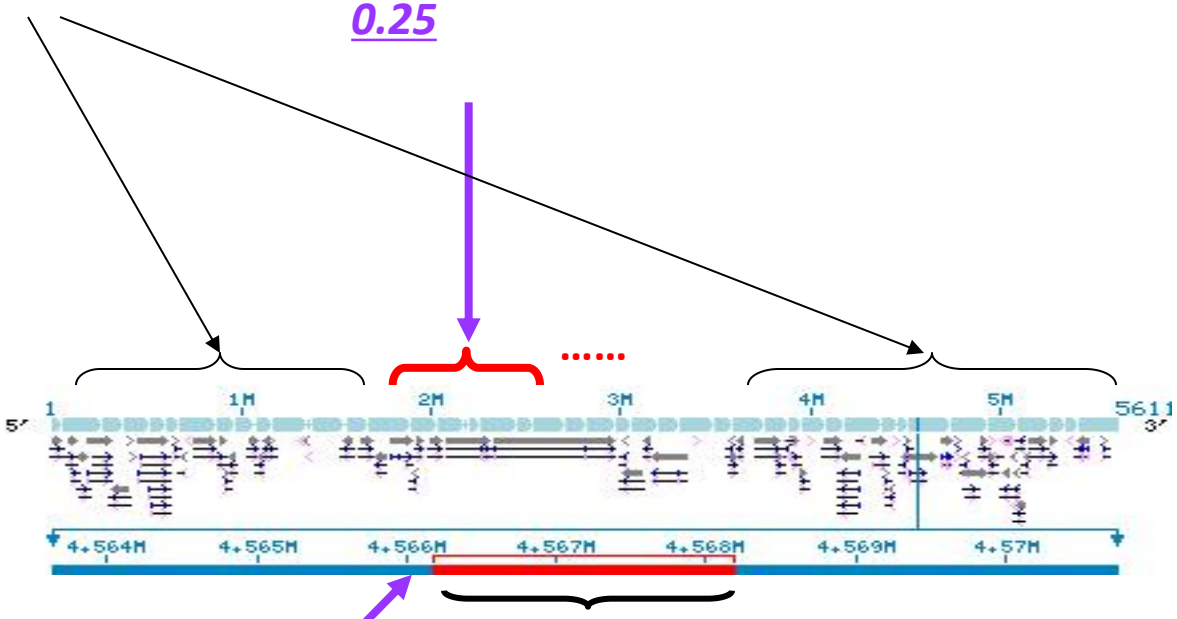
本部分ppt修改自网上资料

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) ~ 0.25



CpG岛

- Away from gene regions:
  - The C in CG pairs is usually *methylated*
  - *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

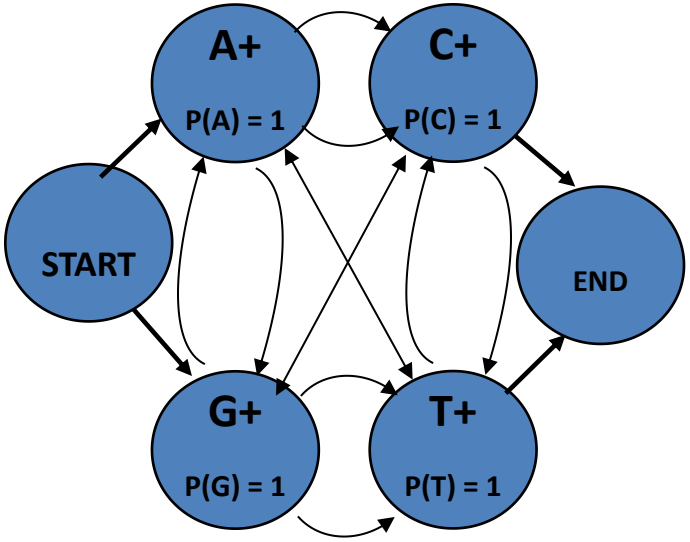
- CpG-rich regions are associated with genes which are *frequently transcribed*.
- Helps to understand gene expression related to *location* in genome.

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

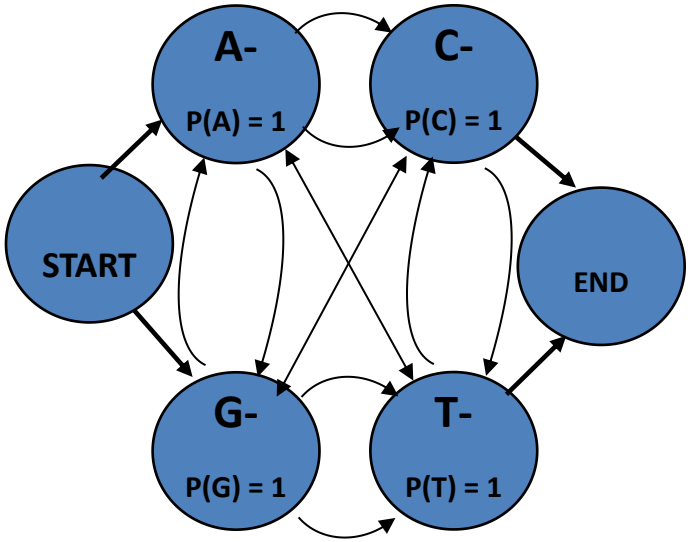
    NON-ISLAND bases …

HMM模型框架

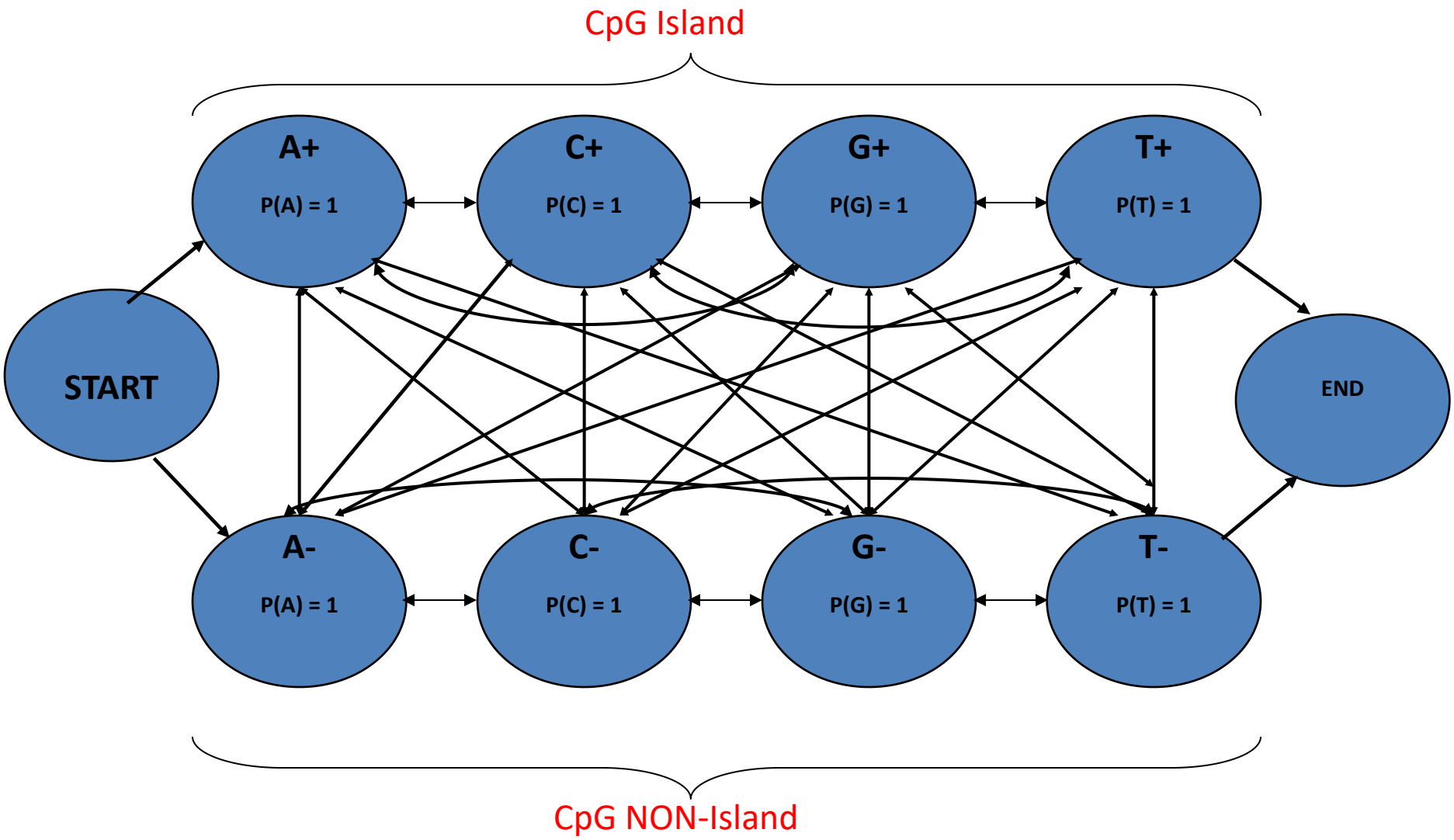


CpG Island (+)

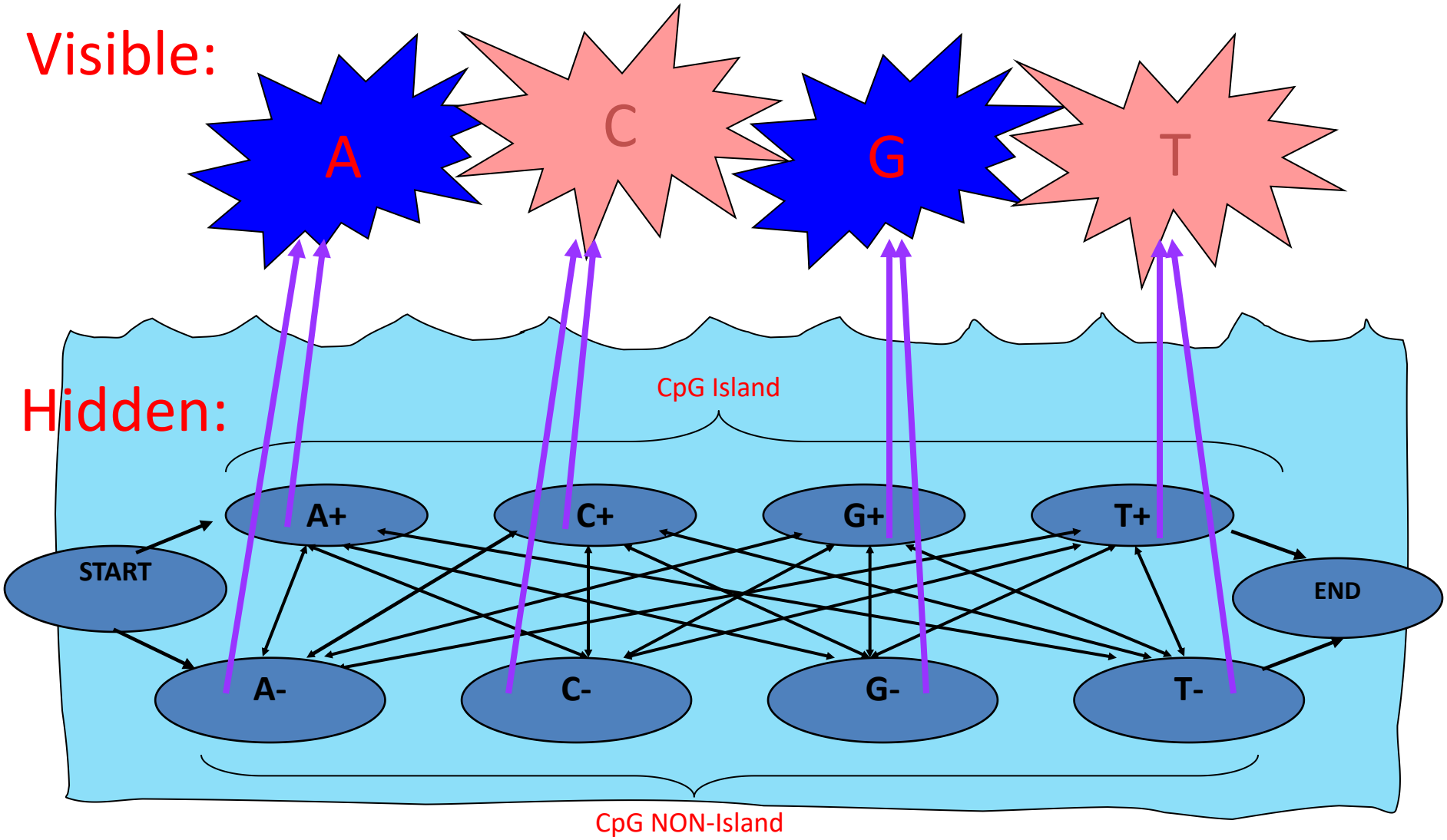
CpG NON-Island (-)



HMM模型框架


CpG Island
A+
P(A) = 1
C+
P(C) = 1
G+
P(G) = 1
T+
P(T) = 1
START
END
A-
P(A) = 1
C-
P(C) = 1
G-
P(G) = 1
T-
P(T) = 1
CpG NON-Island

HMM示意


Visible:
A
C
G
T
Hidden:
CpG Island
A+
C+
G+
T+
START
END
A-
C-
G-
T-
CpG NON-Island

HMM训练问题

CG-RICH sequences

AATAGAGAGGTTCGACTCTGC
ATTTCCCAAATACGTAATGCTT
ACGGTACACGACCCAAGCTCT
CTGCTTGAATCCCAAATCTGA
GCGGACAGATGAGGGGGCG
CAGAGGAAAAACAGGTTTTG
GACCCTACATAAANAGAGAG
GTTCGTAAATAGAGA

HOW?

ML or Forward/ Backward algorithm

CG-POOR sequences

GGTTCGACTCTGCATTTCCCA
AATACGTAATGCTTACGGTTA
AATAGAGAGGTTCGACTCTGC
ATTTCCCAAATACGTAATGCTT
ACGGTACACGACCCAAGCTCT
CTGCTTGTAACTTGTTTTNGT
CGCAGCTGGTCTTGCCTTTGC
TGGGGCTGCTGA

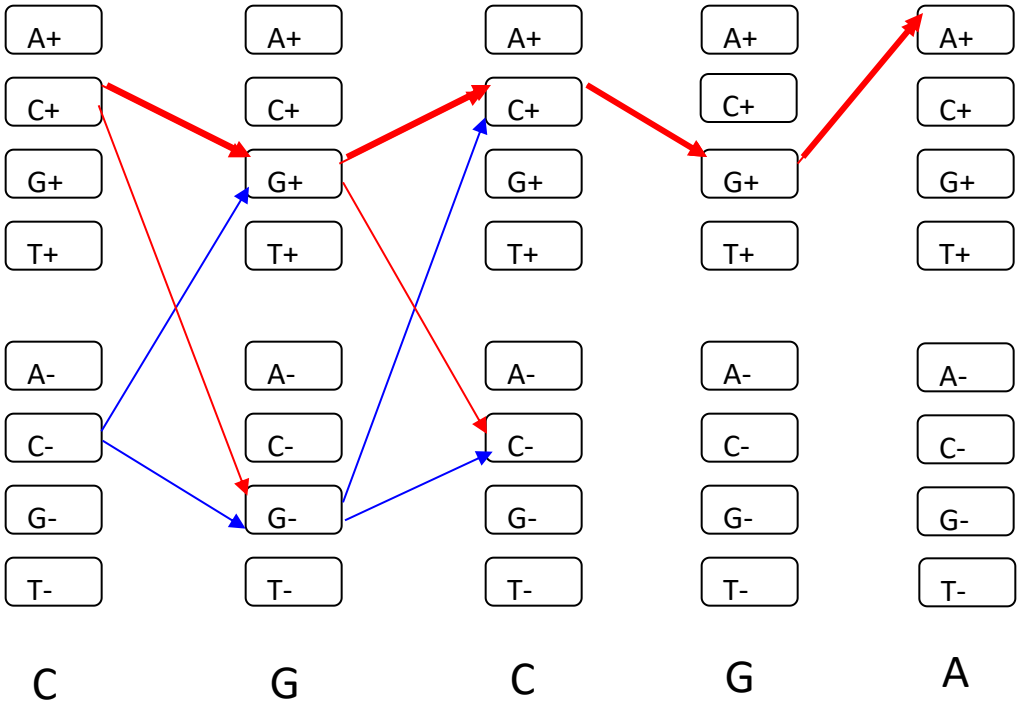
	$A+$	$C+$	$G+$	$T+$	$A-$	$C-$	$G-$	$T-$
$A+$	0.17	0.26	0.42	0.11	0.01	0.01	0.01	0.01
$C+$	0.16	0.36	0.26	0.18	0.01	0.01	0.01	0.01
$G+$	0.15	0.33	0.37	0.11	0.01	0.01	0.01	0.01
$T+$	0.07	0.35	0.37	0.17	0.01	0.01	0.01	0.01
$A-$	0.01	0.01	0.01	0.01	0.29	0.2	0.27	0.2
$C-$	0.01	0.01	0.01	0.01	0.31	0.29	0.07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

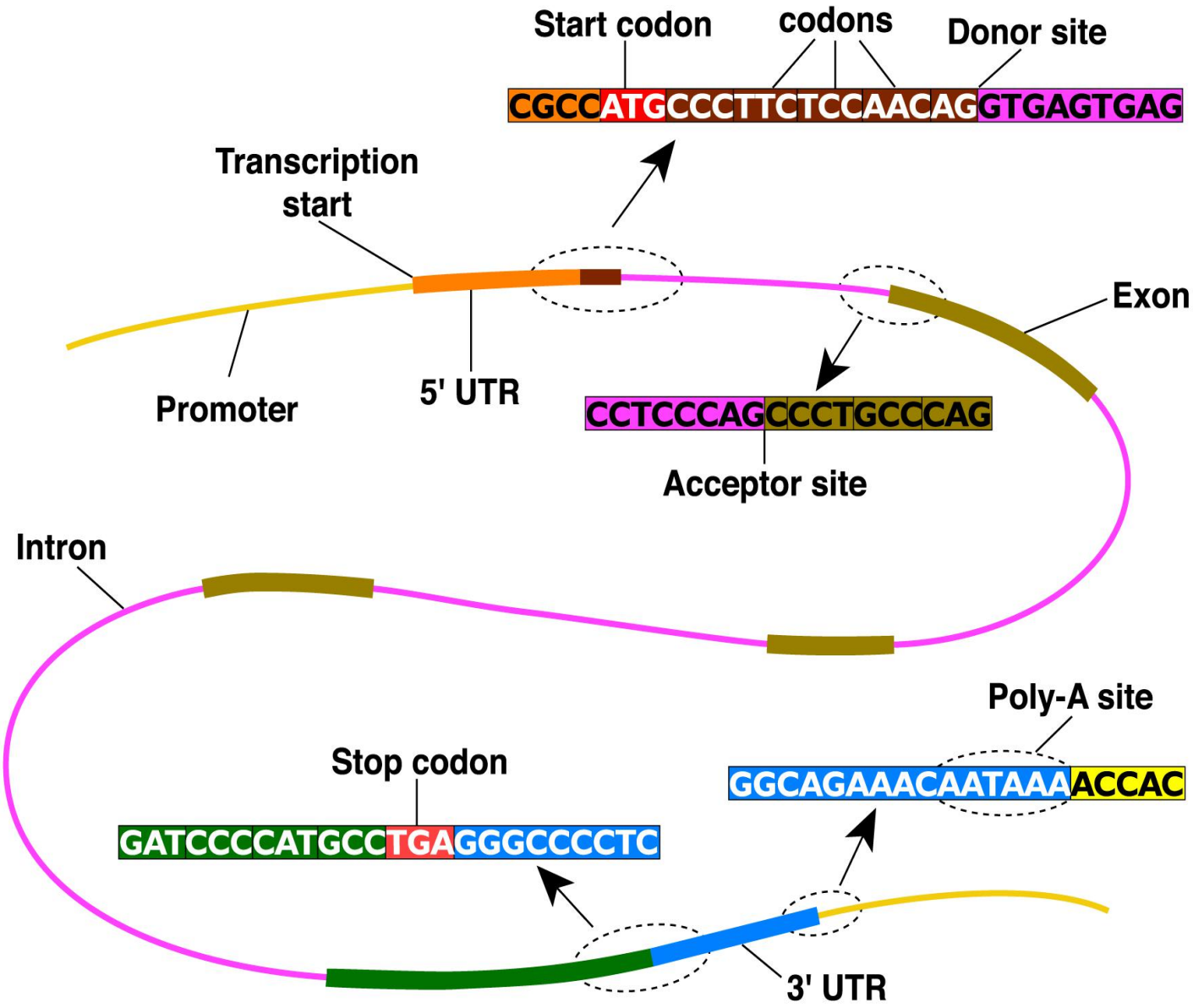
- Decoding- Meaning of observation sequence by looking at the underlying states.
- Hidden states A+,C+,G+,T+,A-,C-,G-,T-
- Observation sequence CGCGA
- State sequences C+,G+,C+,G+,A+ or C-,G-,C-,G-,A-
  or C+,G-,C+,G-,A+
- Most Probable Path C+,G+,C+,G+,A+

HMM解码问题

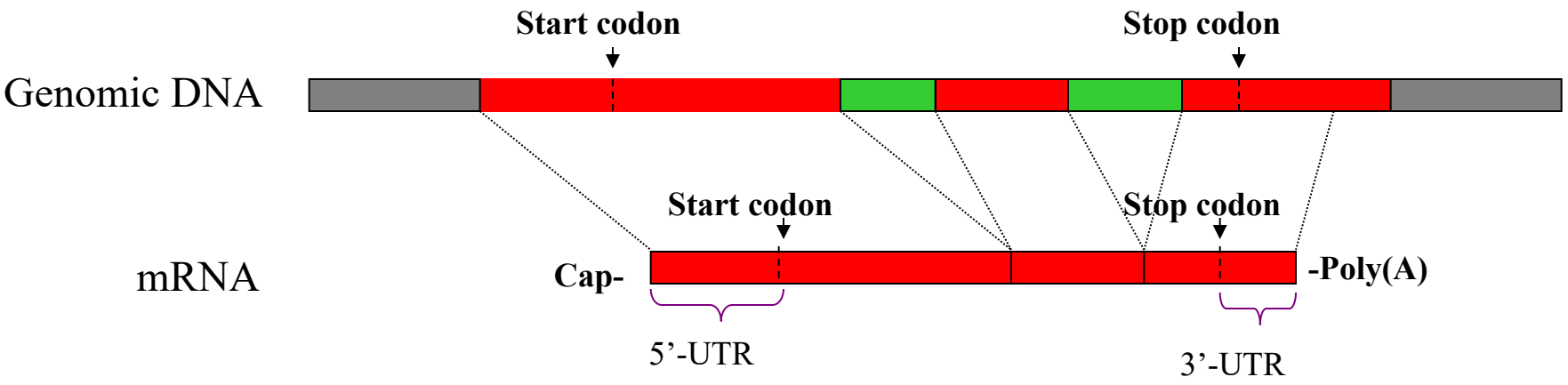


应用II：Gene Finding

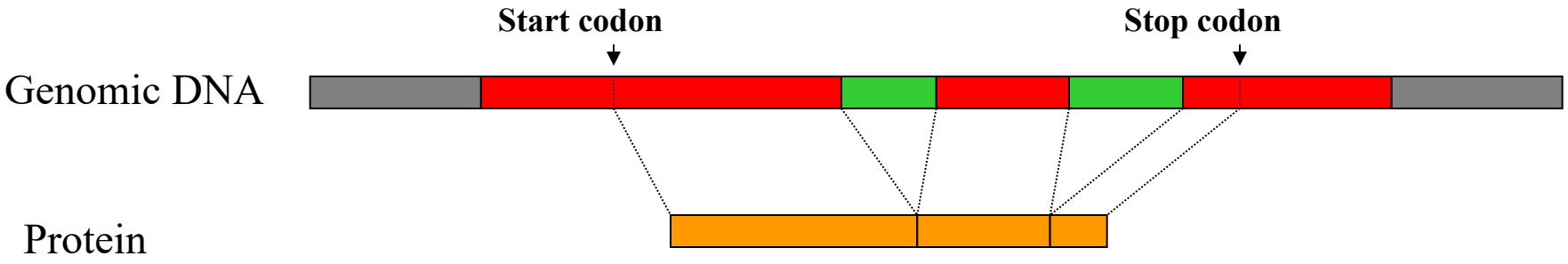
- *This part is modified from slides download from www.cs.ubc.ca/~rogic/**GeneFinding.ppt***


Start codon
codons
Donor site
CGCCATGCCCTTCTCCAACAGGTGAGTGAG
Transcription start
Exon
Promoter
5' UTR
CCTCCCAGCCCTGCCCAG
Acceptor site
Intron
Poly-A site
Stop codon
GGCAGAAACAATAAAACCAC
GATCCCCATGCCTGAGGGCCCCTC
3' UTR

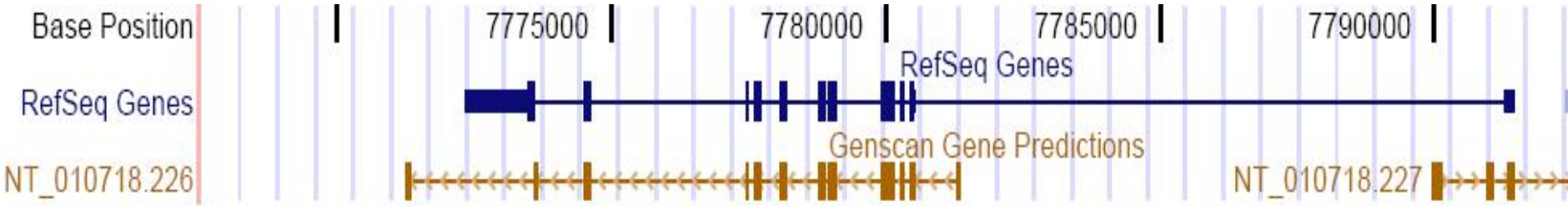
Spliced Alignment
Compare with cDNA or EST probes



Spliced Alignment
Compare with Protein Probes

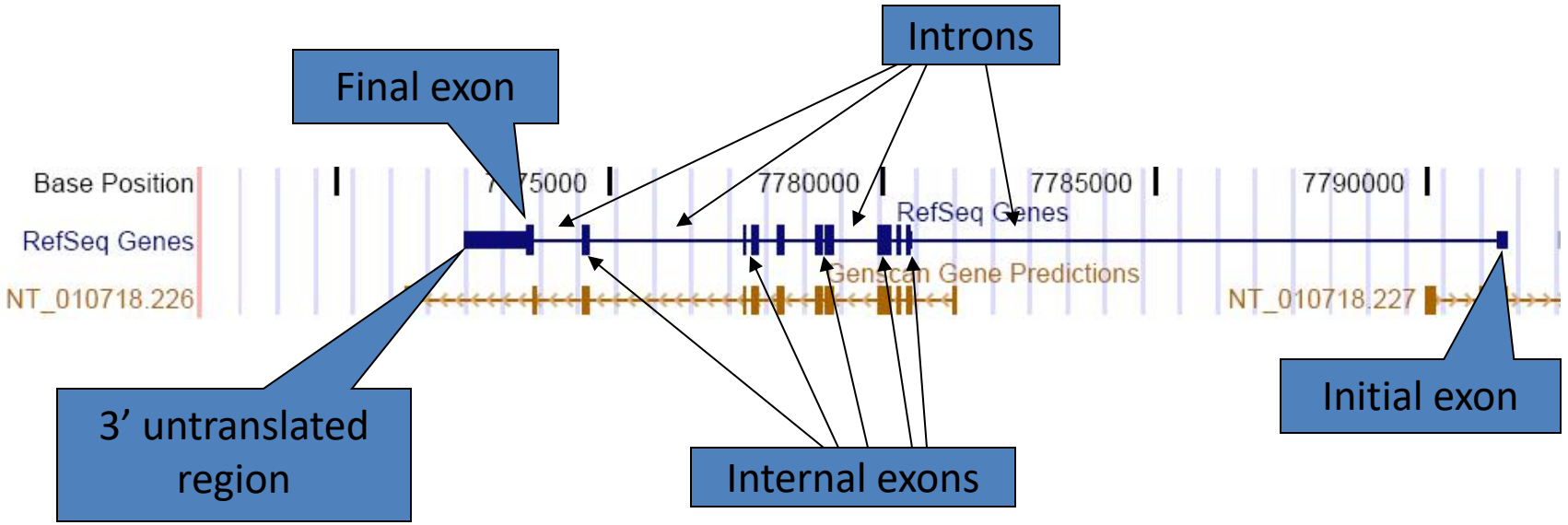


A Eukaryotic Gene



- 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.

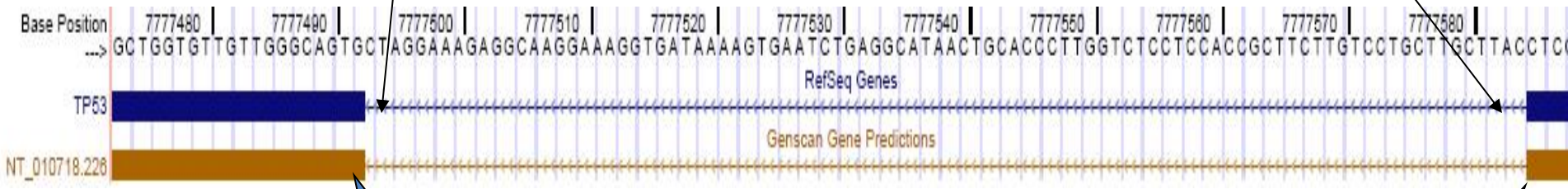
An Intron

revcomp(CT)=AG

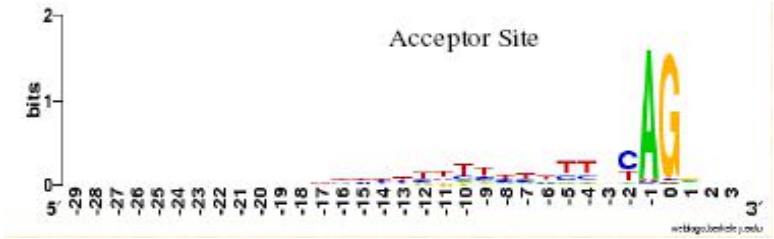
GT: signals start of intron

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 ~ 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 ~50%

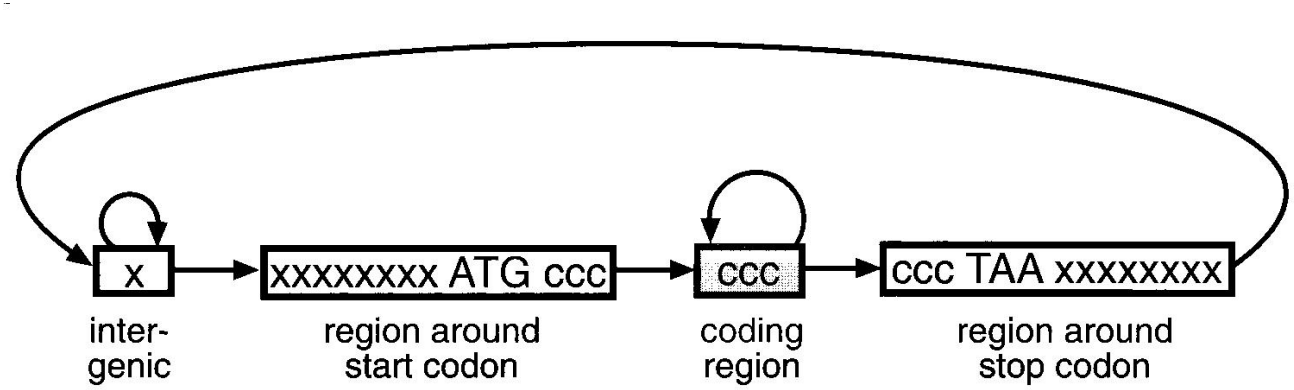
Problems:

- many

HMMs and Gene Structure

- Nucleotides *{A,C,G,T}* are the observables
- Different states generates 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

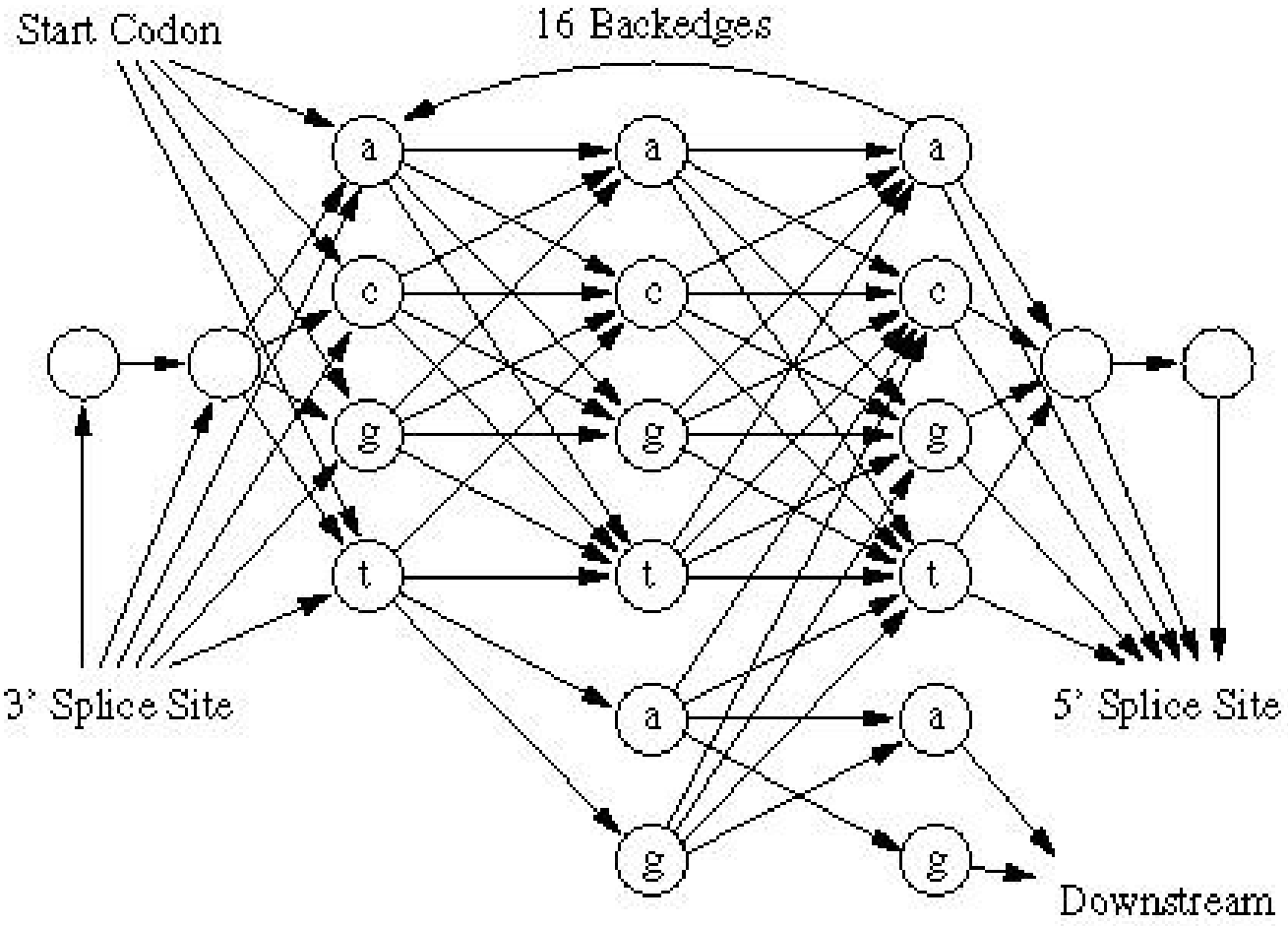
Examples of Gene Finders Using HMM

- 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**

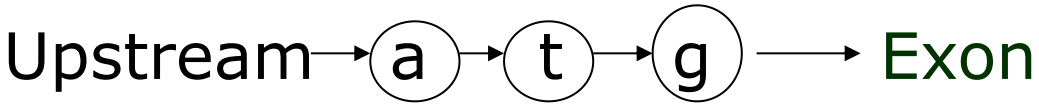
- The **Viterbi Exon-Intron Locator** (**VEIL**) was developed by John Henderson, Steven Salzberg, and Ken Fasman at Johns Hopkins University.
- Gene finder with a modular structure:
- Uses a HMM which is made up of sub-HMMs each to describe a different bit of the sequence: upstream noncoding DNA, exon,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

The Exon Sub-model

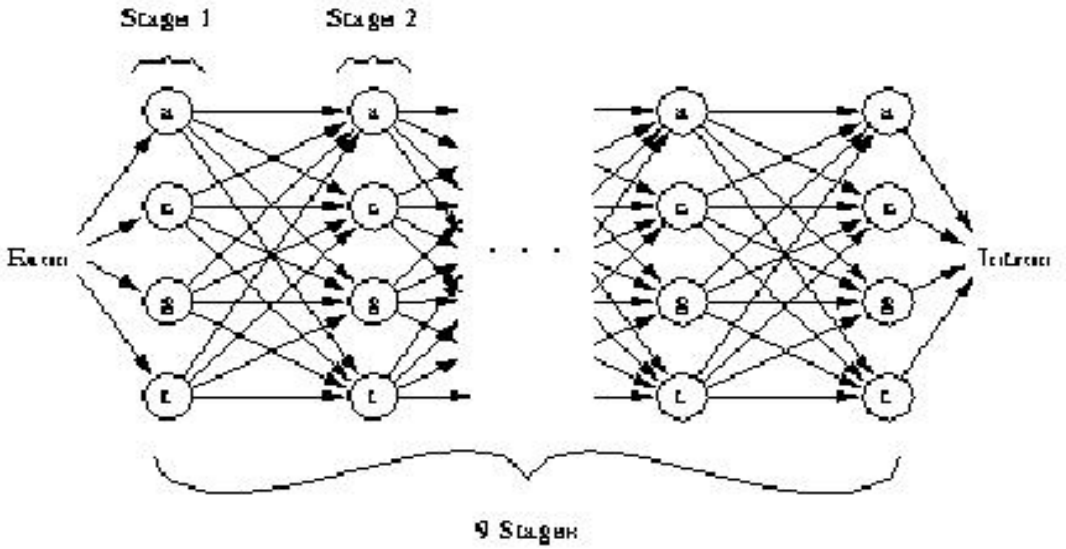


Other Submodels

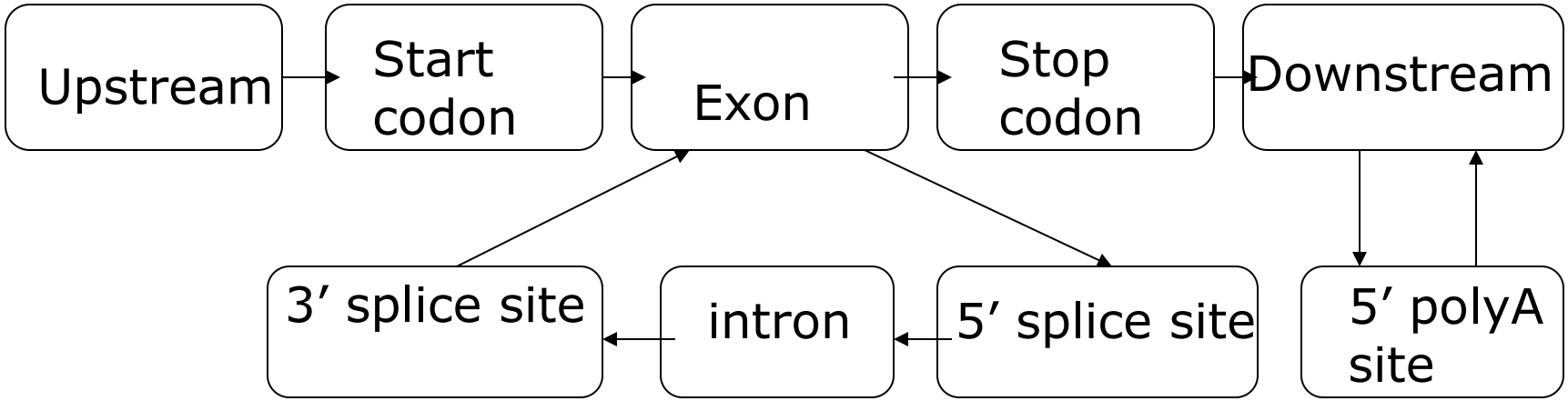
- The start codon model is very simple:



- The splice junctions are also quite simple and can be hardwired (here is the 5’ splice site):



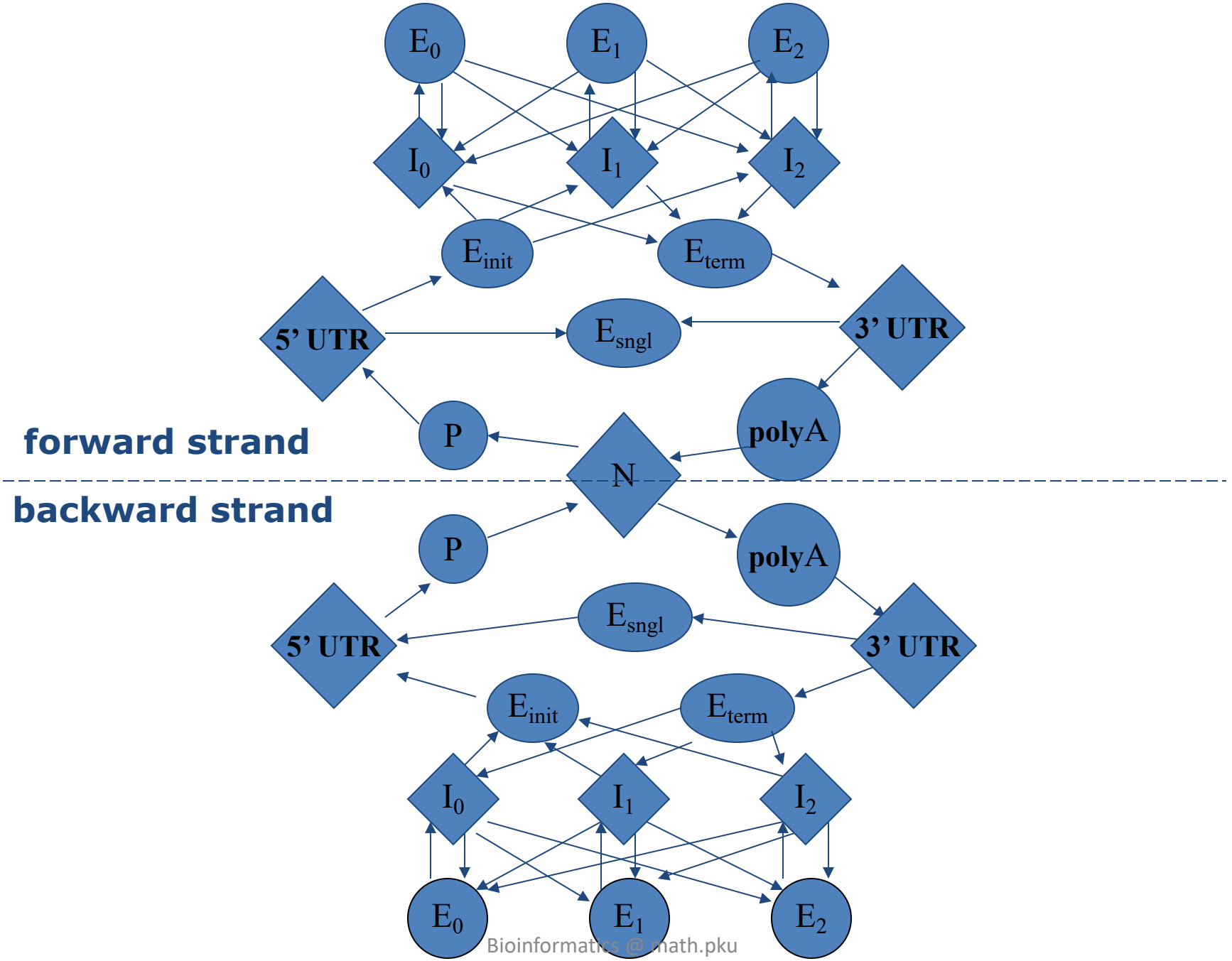
The Overall Model



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 to model gene structure (different length distributions for exons)
- Different model parameters for regions with different GC content



Genscan's Architecture

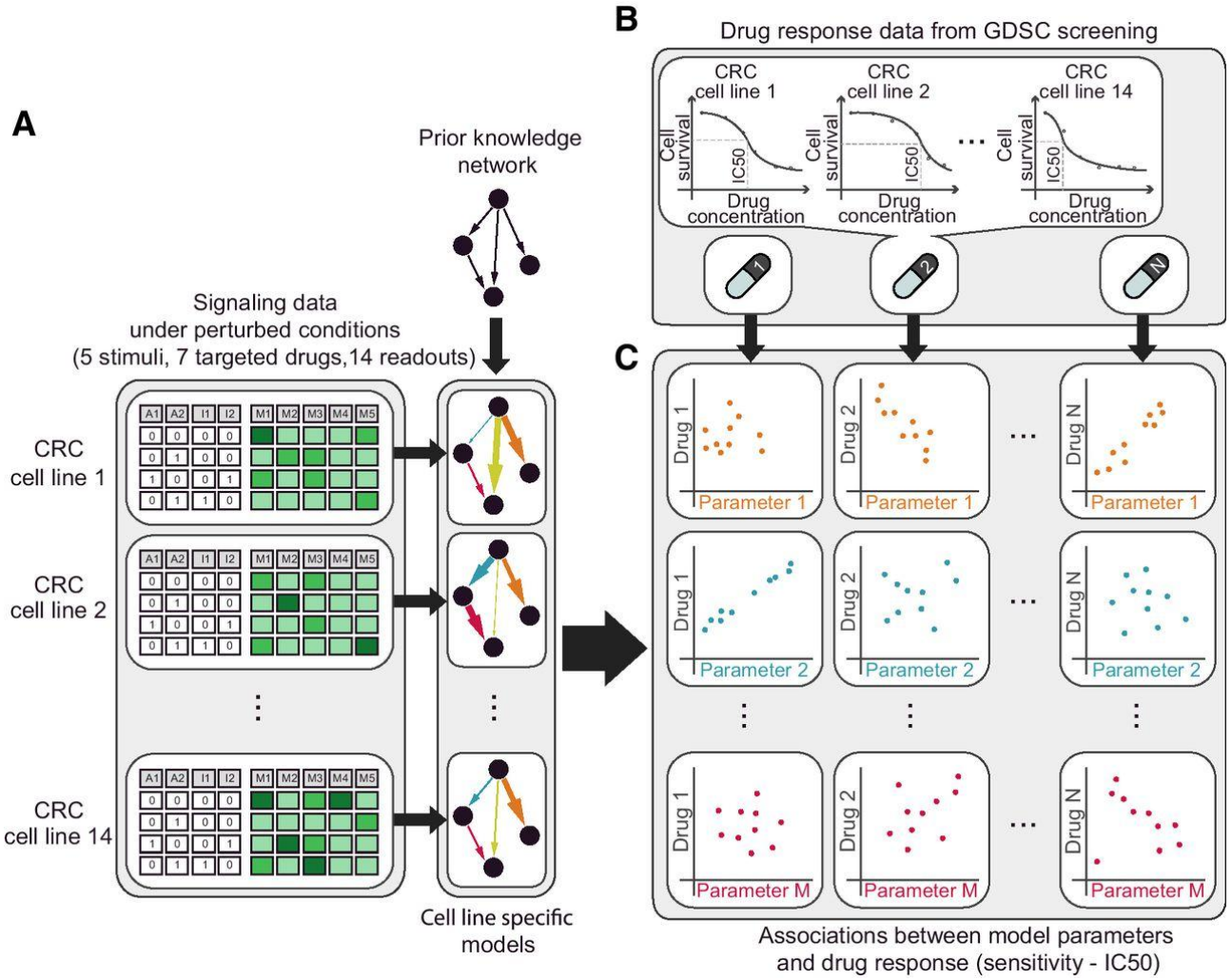
- HMM's states for exons and introns in three different phases, single exon, 5' and 3' UTRs, promoter region and polyA site and intergenic region
- Explicit length modeling
- HMMs for exons, introns and intergenic regions
- Weight matrix (WM) and weight array(WA) for acceptor site, branch point, polyA site and promoter region
- Decision tree (maximal dependence decomposition) for donor sites

For more detail, see:

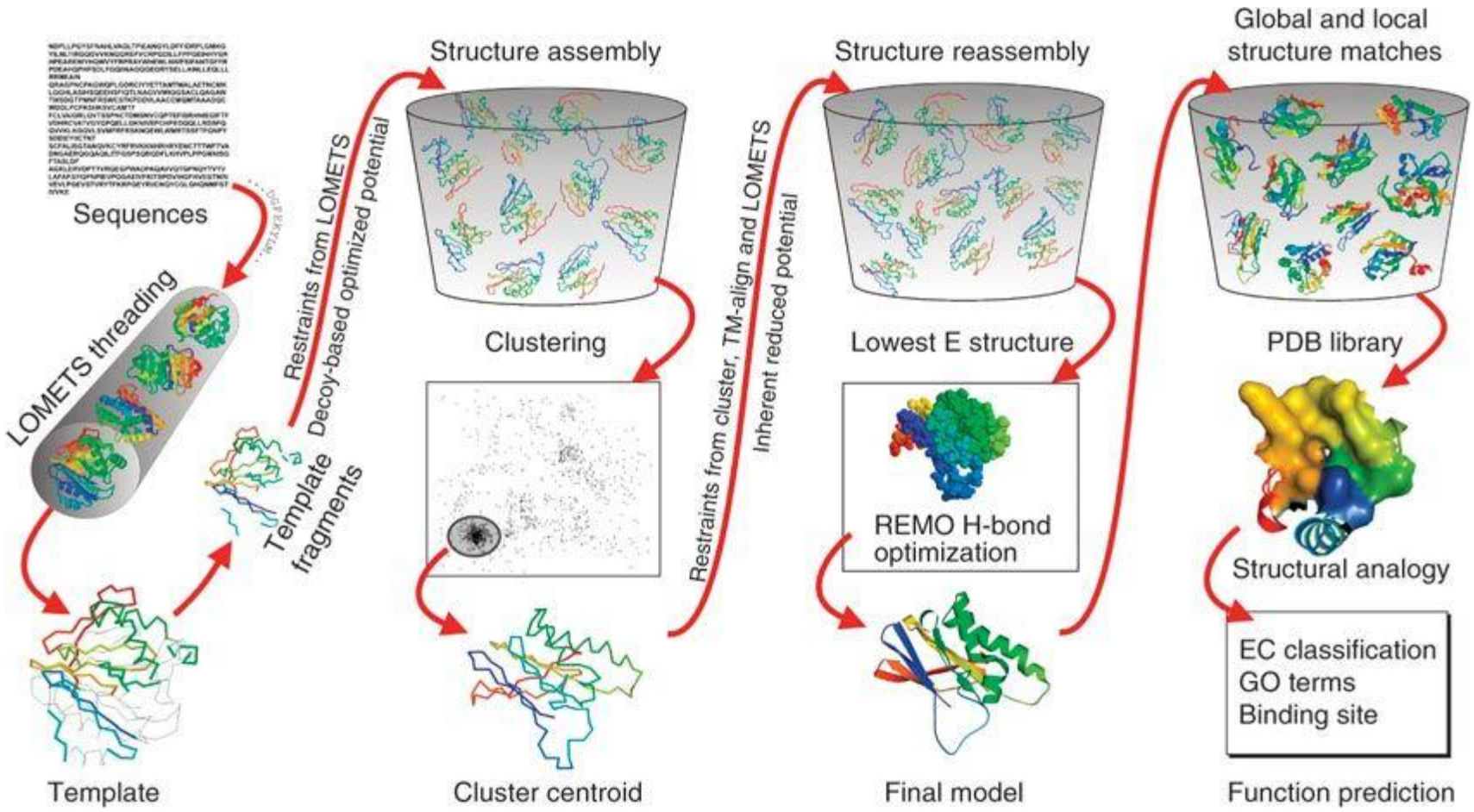
Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* **268,** 78-94

Burge, C. B. and Karlin, S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.* **8,** 346-354.

思考：HMM只能应用在序列上吗？



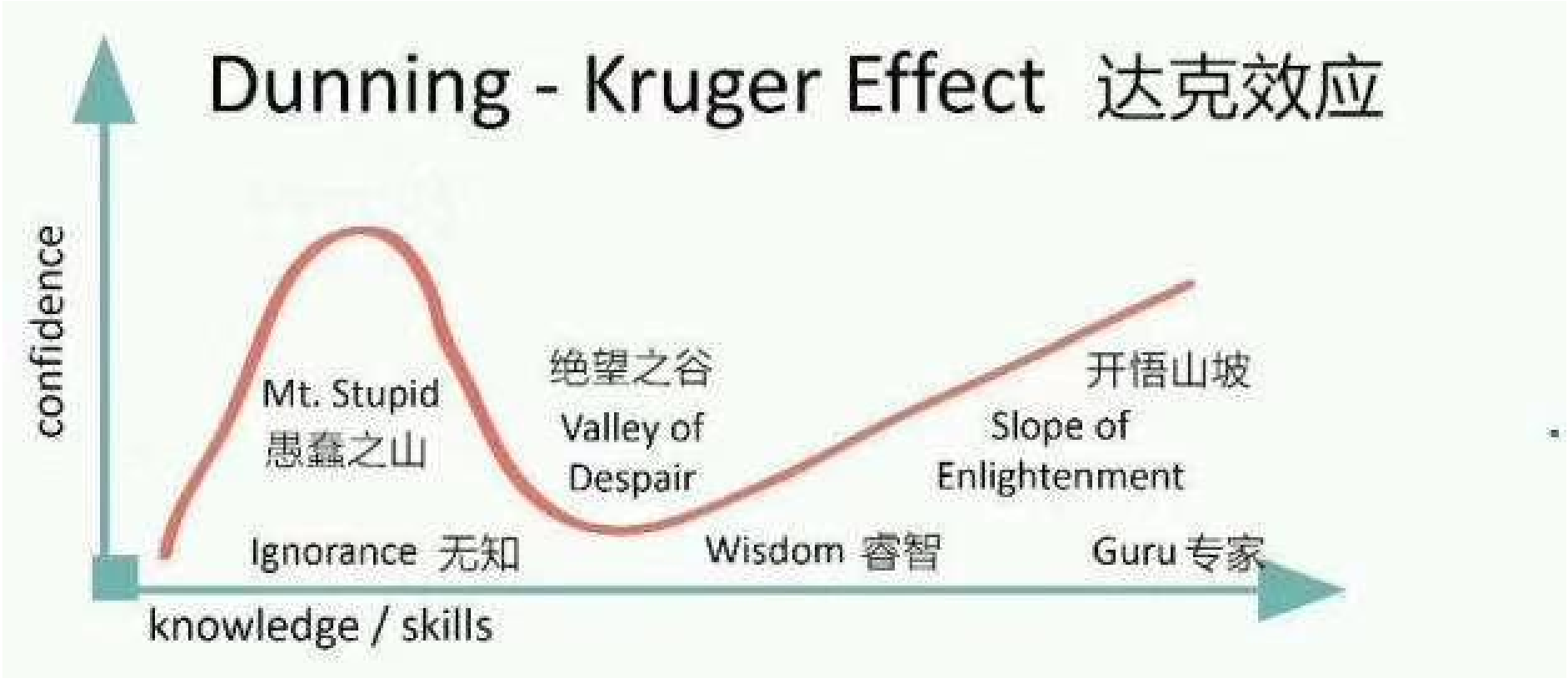
思考：HMM只能应用在序列上吗？



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.

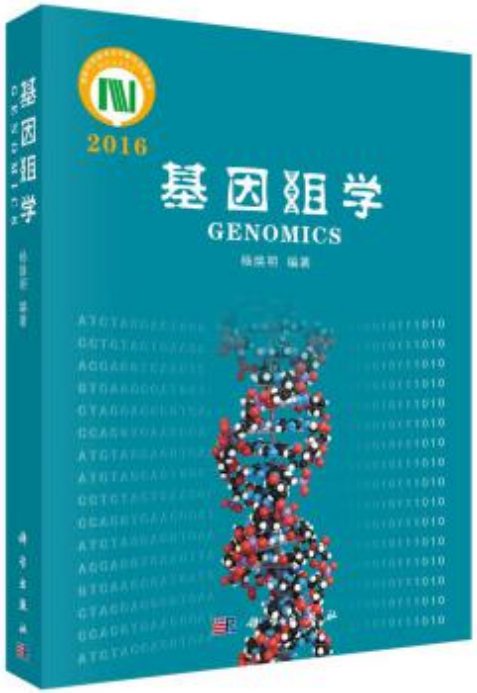
我们学会了HMM算法！？

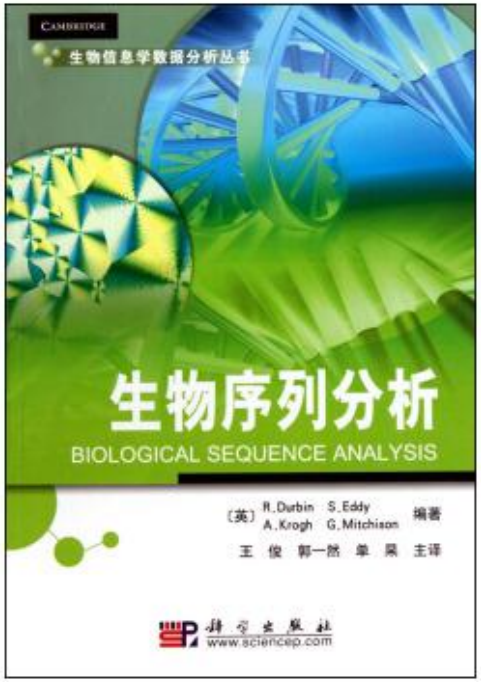

Dunning - Kruger Effect 达克效应
confidence
Mt. Stupid
愚蠢之山
绝望之谷
Valley of
Despair
开悟山坡
Slope of
Enlightenment
Ignorance 无知
Wisdom 睿智
Guru 专家
knowledge / skills

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

References


2016
基因组学
GENOMICS


CAMBRIDGE
生物信息学数据分析丛书
生物序列分析
BIOLOGICAL SEQUENCE ANALYSIS
R.Durbin S.Eddy
A.Krogh G.Mitchison
编著
王俊 郭一然 单昊 主译
www.sciencep.com


生物统计学
(第五版)

生物统计经典软件

基因组可视化：Genome Browser，(http://genome.ucsc.edu/), (tracks, annotations, etc.)

序列保守性：WebLogo，(http://weblogo.berkeley.edu/logo.cgi),

基因预测：MEME, (http://meme-suite.org/).

进化树：iTOL, (https://itol.embl.de/),

基因调控网络： GeneNetwork, (http://gn2.genenetwork.org/), Cytoscape, (https://cytoscape.org/),

代谢通路：KEGG, (https://www.kegg.jp/); iPATH, (https://pathways.embl.de/),

蛋白结构与功能：PDB，(http://www.rcsb.org); pFAM, (http://pfam.xfam.org/),

微生物组: EBI Magnify. (https://www.ebi.ac.uk/metagenomics/),

蛋白和小分子互作数据：STITCH，(http://stitch.embl.de/); STRING, (http://string-db.org),

药物数据库：DrugBank, (https://www.drugbank.ca/),

生物数据分析平台：Galaxy, (https://usegalaxy.org/),

生物数据可视化：Echart, (https://www.echartsjs.com/examples/zh/index.html),