中国科学院大学
University of Chinese Academy of Sciences

自然语言处理
Natural language Processing

授课教师：胡玥

2024.12

课程编码： 1800860812O3P2002H 课程名称：自然语言处理 授课团队 黄河燕、胡玥 、张仰森

第 9 章　NLP基础任务

序列标注

2/3

内 容 提 要

9.0　概述

9.1　文本分类

9.2　文本匹配

9.3　序列标注

9.4　序列生成

■ 序列标注

本节内容：

1. 序列标注问题概述
2. 隐马尔科夫模型HHM(概率统计模型)
4. 神经网络序列标注模型（深度学习模型）

1. 序列标注问题概述

问题引入

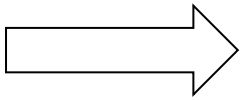
- **实体识别与抽取** （在给定的序列中找到目标序列）

非结构化文本

2011年7月25日，在上海举办的游泳世锦赛上，年仅15岁的叶诗文的以2分08秒90的成绩勇夺女子200米混合泳冠军，成为最年轻的单项世界冠军获得者。

实体抽取

时间/地名/人名/...



2011年7月25日，在上海举办的游泳世锦赛上，年仅15岁的叶诗文的以2分08秒90的成绩勇夺女子200米混合泳冠军，成为最年轻的单项世界冠军获得者。

标注任务	输入	任务建模	输出
	非结构化文本序列	序列标注模型	目标片段 一般用序列标注方法

1. 序列标注问题概述

■ **例1：实体识别类问题**

问题1：将给定的输入序列中的人名识别出来（人名识别）

新任总裁罗建国宣布了对部门经理邓奇的任免通知

问题2：将给定的输入序列中的组织机构名识别出来（组织机构名）

新任总裁罗建国宣布了对远大公司经理国庆的任免通知

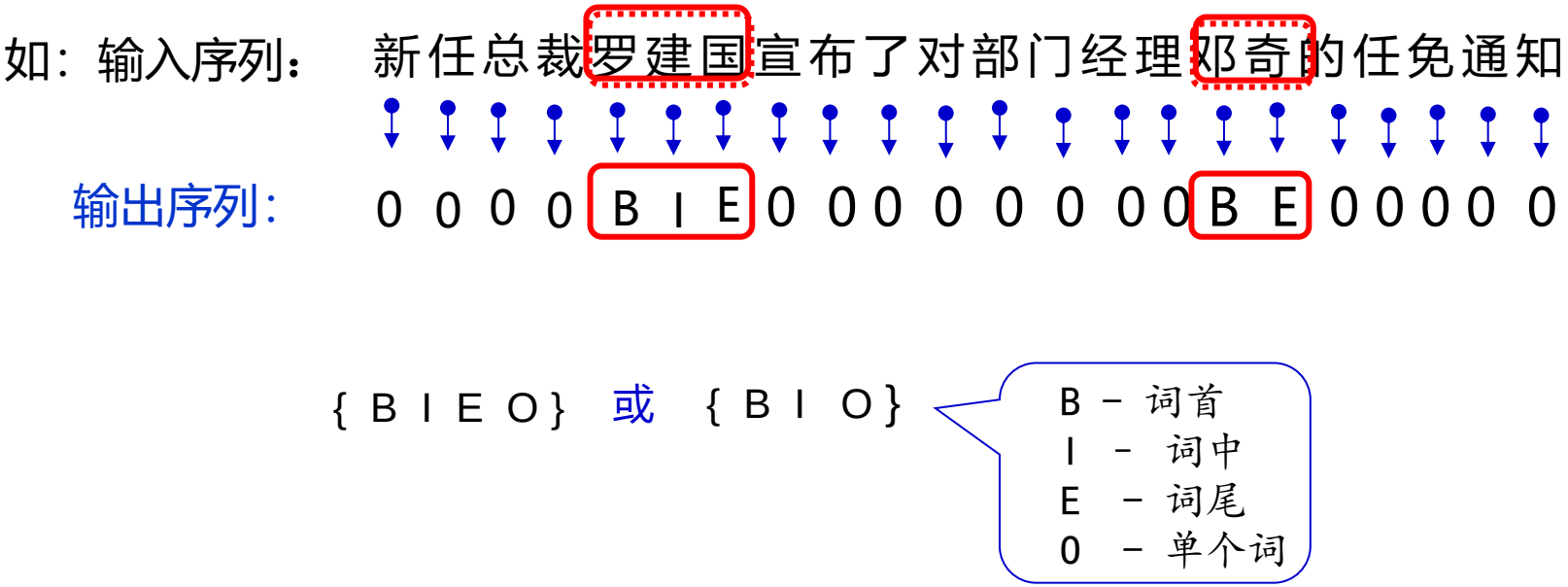
问题3：将给定的输入序列中的军事术语抽取出来

鹰 式 战 斗 机 是 一 款 极 为 优 秀 的 多 用 途 战 斗 机

1. 序列标注问题概述

序列标注建模：“将输入的语言序列转化为标注序列”，通过标注序列标签含义来解决问题。如：

◆ **命名实体识别（人名识别）**



1. 序列标注问题概述

◆ **命名实体识别（组织机构名识别）**

输入序列：新任总裁罗建国宣布了对远大公司经理国庆的任免通知

标注序列：O OO OOOO O OOO BIIEOOO OOOO OO

◆ **信息抽取（实体识别）**

输入：鹰 式 战 斗 机 是 一 款 极 为 优 秀 的 多 用 途 战 斗 机

输出：B I I I E O O O O O O O O B I I I I E

1. 序列标注问题概述

PROB (*the* | ART)	0.54
PROB (*flies* | N)	0.025
PROB (*flies* | V)	0.076
PROB (*like* | V)	0.1
PROB (*like* | P)	0.068
PROB (*like* | N)	0.012
PROB (*a* | ART)	0.360
PROB (*a* | N)	0.001
PROB (*flower* | N)	0.063
PROB (*flower* | V)	0.05
PROB (*birds* | N)	0.076

- **例2：词性标注（POS）类问题**

概率统计NLP中的词法分析中词性标注

目标：将给定的输入序列中词的词性标出来

如：　　输入：　Flies　like　a　flower

词性标注结果：　Flies/N　like /V　a/ART　flower/N

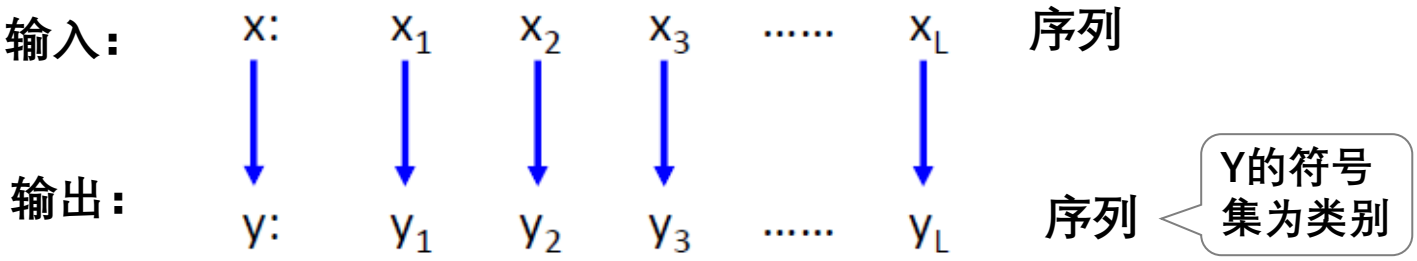
标签集合集　{ 单词的词性，如 N 、V 等 }

1. 序列标注问题概述

序列标注方法：

$$f: \quad X \rightarrow Y$$

序列　　序列



标注问题是分类问题的推广，是复杂结构预测的简单形式（监督学习）

许多自然语言处理问题 均可转化为序列标注问题，如 分词、短语识别、依存分析、语义角色标注，信息抽取……

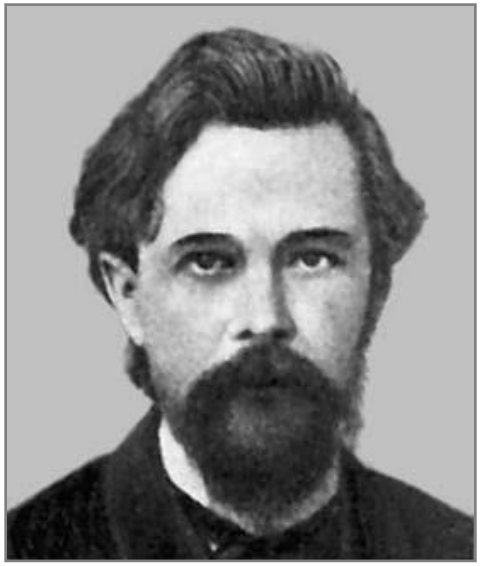
■ 序列标注

本节内容：

1. 序列标注问题概述
2. 隐马尔科夫模型HHM(概率统计模型)
   - 2.1 马尔科夫模型
   - 2.2.隐马尔科夫模型
3. 神经网络序列标注模型（深度学习模型）

2.1 马尔科夫模型

马尔可夫（Andrei Andreyevich Markov, 18510.6.14 ~ 1922.7.20 ）

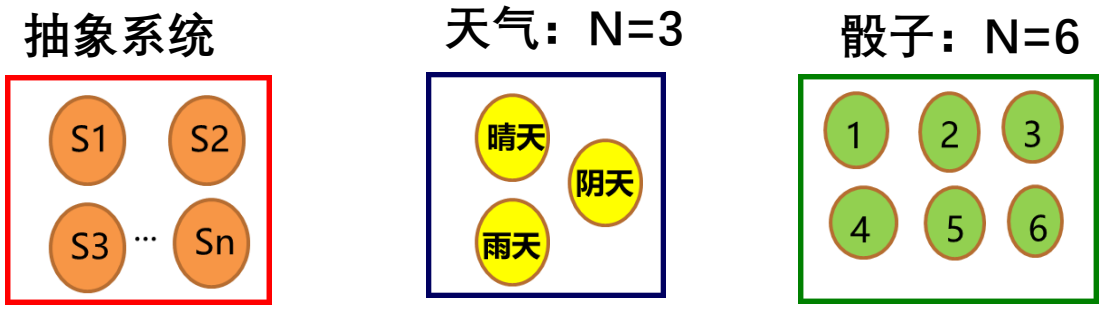


前苏联数学家。切比雪夫(1821年5月16日 ~ 1894年12月8日)的学生。在概率论、数论、函数逼近论和微分方程等方面卓有成就。他提出了用数学分析方法研究自然过程的一般图式—马尔可夫链，并开创了随机过程(马尔可夫过程) 的研究。

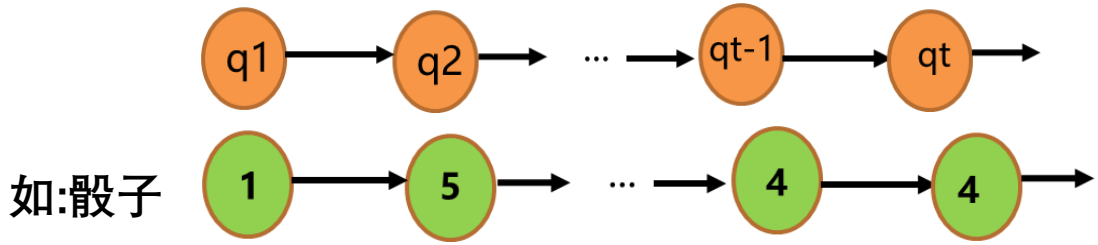
2.1 马尔科夫模型

马尔可夫链

一个系统有N个状态 S1，S2，…，Sn，



随着时间（空间）推移，系统从某一状态转移到另一状态



如果系统在t时间的状态q t 只与其在时间 t -1的状态相关[$P(q_i|q_{i-1}\cdots q_1)$ = $P(q_i|q_{i-1})$]，则系统构成离散的一阶马尔可夫链(马尔可夫过程)

2.1 马尔科夫模型

马尔可夫模型(马尔可夫链出现的概率)：

$$p(S_0, S_1, \ldots, S_T) = \prod_{t=1}^{T} p(S_t|S_{t-1})\ p(S_0)$$

模型输入： 状态序列

模型输出： 状态序列的概率值

模型参数： $P(q_t|q_{t-1})$)

	s1	s2	s3	sn
s1	a11	a12	a13 ·····	a1n
s2	a21	a22	a23 ·····	a2n
s3				
⋮	⋮			
sn	an1	an2	an3 ·····	anr

独立于时间 t 的随机过程：

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{i,j}, 1 \le i, j \le N$$

其中：状态转移概率 a_{ij} 必须满足 $a_{ij}>=0$ ，

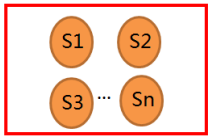
且 $\sum_{j=1}^{N} a_{i,j} = 1$

2.1 马尔科夫模型

马尔可夫模型组成

三元组　M = （S, π，A）

S={ s1　s2　s3 …..　sn　}



A=[aij]=

	s1	s2	s3	sn
s1	a11	a12	a13 ·····	a1n
s2	a21	a22	a23 ·····	a2n
s3				
⋮	⋮			
sn	an1	an2	an3 ·····	anr

其中： 状态转移概率 a_{ij}

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{i,j}, 1 \le i, j \le N$$

满足 $a_{ij} >= 0$，且 $\sum_{j=1}^{N} a_{i,j} = 1$

π　初始状态向量

参数	含义
S	模型中状态的有限集合
A	与时间无关的状态转移概率矩阵
π	初始状态空间的概率分布

2.1 马尔科夫模型

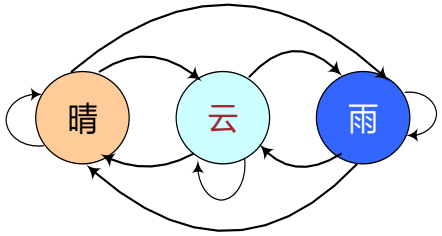
马尔可夫模型作用

三元组　M = （S, π，A）

参数	含义
S	模型中状态的有限集合
A	与时间无关的状态转移概率矩阵
π	初始状态空间的概率分布

定量描述随机事件：天气变化

例1：预测天气变化



S={ 晴　云　雨 }

A

	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

π　=(1, 0, 0)

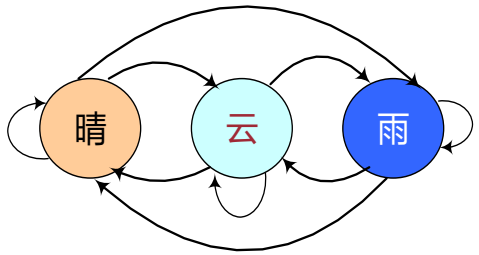
2.1 马尔科夫模型

假定一段时间内的气象可由一3状态马尔可夫模型 M 描述：

S_1：雨，S_2：多云，S_3：晴，转移概率矩阵为：

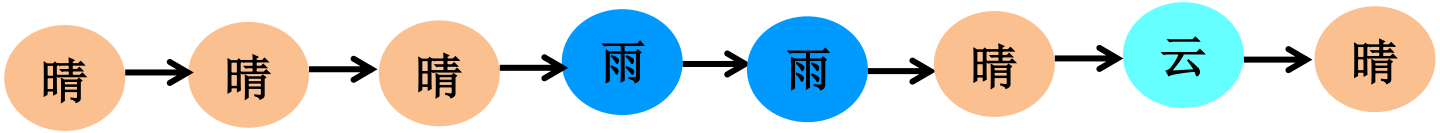
$A = [a_{ij}] =$

	S_1	S_2	S_3
S_1	0.4	0.3	0.3
S_2	0.2	0.6	0.2
S_3	0.1	0.1	0.8



如果第一天为晴天，根据这一模型，求在今后七天中天气为 S=“晴晴雨雨晴云晴”的概率

即，求



的概率

2.1 马尔科夫模型

解：　马尔可夫模型状态序列概率：

$$p(S_0, S_1, \ldots, S_T) = \prod_{t=1}^{T} p(S_t|S_{t-1})\ p(S_0)$$

A = [a_{ij}]=

	雨	云	晴
雨	0.4	0.3	0.3
云	0.2	0.6	0.2
晴	0.1	0.1	0.8

S = 晴 晴 晴 雨 雨 晴 云 晴

$$\begin{aligned}
&P(O \mid M) \\
&= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid M) \\
&= P(S_3) \cdot P(S_3 \mid S_3) \cdot P(S_3 \mid S_3) \cdot P(S_1 \mid S_3) \cdot P(S_1 \mid S_1) \cdot \\
&P(S_3 \mid S_1) \cdot P(S_2 \mid S_3) \cdot P(S_3 \mid S_2) \\
&= 1 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\
&= (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}$$

■ 序列标注

本节内容：

1. 序列标注问题概述
2. 隐马尔科夫模型HHM(概率统计模型)
   - 2.1 马尔科夫模型
   - 2.2.隐马尔科夫模型
3. 神经网络序列标注模型（深度学习模型）

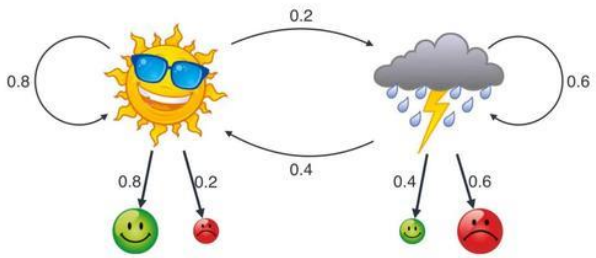
2.2 隐马尔科夫模型

隐马尔可夫模型 （Hidden Markov Model, HMM）

描写：该模型是一个双重随机过程，我们不知道具体的状态序列，只知道状态转移的概率，即模型的状态转换过程是不可观察的（隐蔽的）而可观察事件的随机过程是隐蔽状态转换过程的随机函数。

--- 创建于20世纪70年代 ---

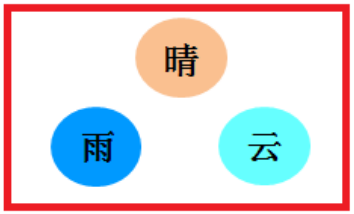
通过可见的事物的变化揭示深藏其后的内在的本质规律



2.2 隐马尔科夫模型

马尔可夫模型：

S:



A:

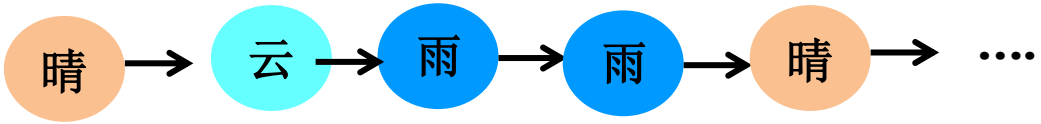
$A = [a_{ij}] =$

	雨	云	晴
雨	0.4	0.3	0.3
云	0.2	0.6	0.2
晴	0.1	0.1	0.8

π：晴 云 雨

(1，0，0)

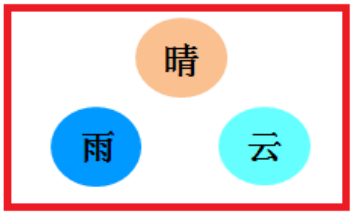
天气变化



2.2 隐马尔科夫模型

隐马尔可夫模型HMM：

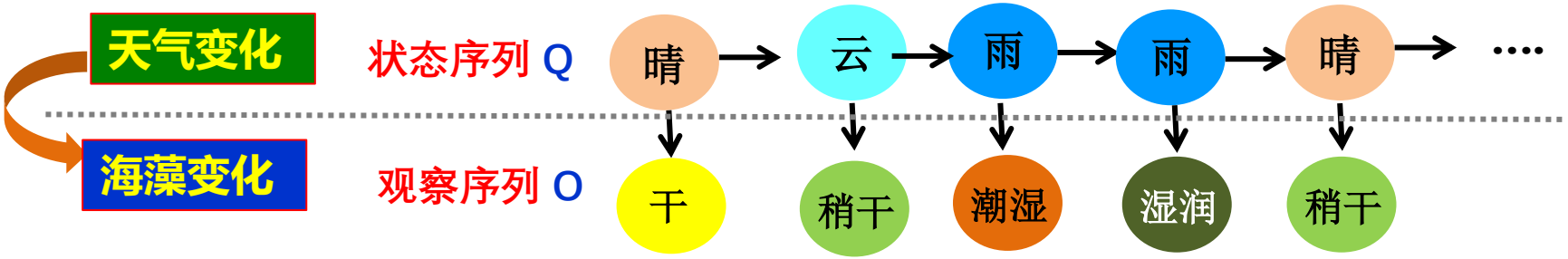
S:



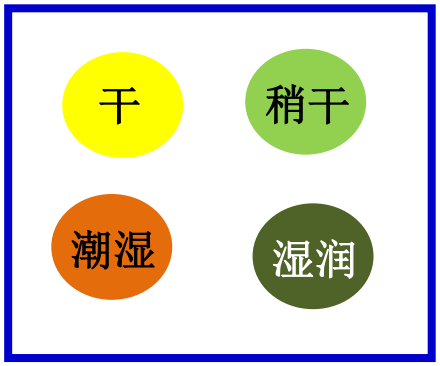
A:

A = [a_{ij}]=	雨	云	晴
雨	0.4	0.3	0.3
云	0.2	0.6	0.2
晴	0.1	0.1	0.8

π：晴 云 雨
(1, 0, 0)



O:



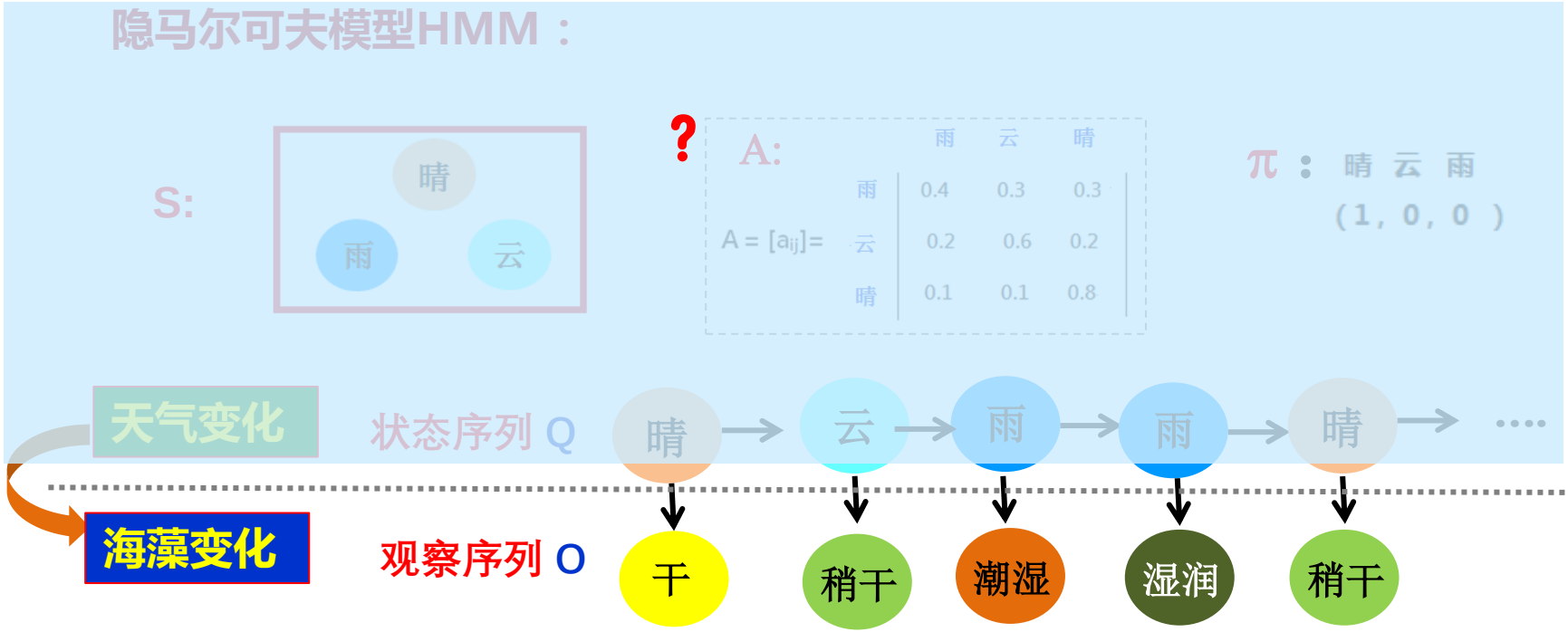
B:

天气 \ 海藻	干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

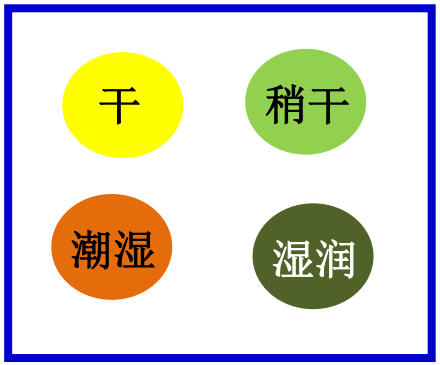
观察序列变化由状态序列变化引起

（两者相关联）

2.2 隐马尔科夫模型



O:



B:

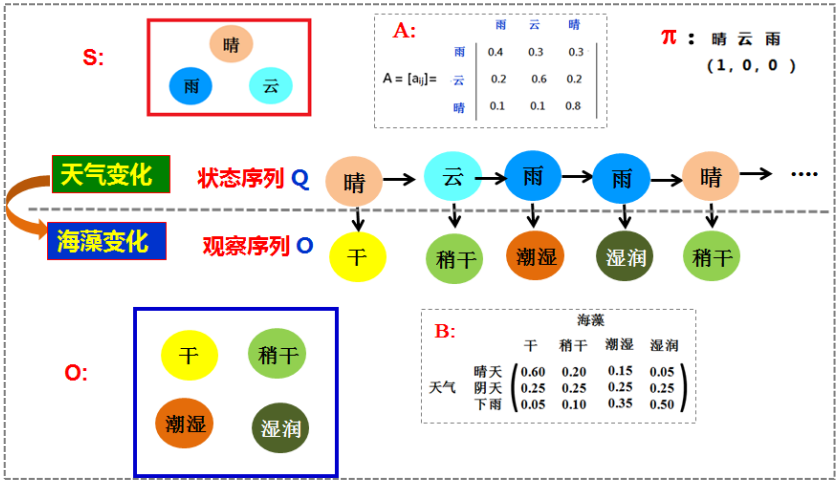
天气	海藻：干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

观察序列变化由状态序列变化引起

（两者相关联）

2.2 隐马尔科夫模型

隐马尔可夫模型(HMM):



要素	含义	实例
S	模型中状态的有限集合	天气
O	每个状态可能的观察值	海藻
A	与时间无关的状态转移概率矩阵	天气转移概率矩阵
B	给定状态下，观察值概率分布	每个天气状态的海藻观测概率
π	初始状态空间的概率分布	初始时选择某天气概率

五元组 $\lambda = (S, O, \pi, A, B)$

或简写为 $\lambda = (\pi, A, B)$

HMM的特点：

- 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。
  - 一般随机函数：描述状态与观察序列间的关系， 用观察值概率描述。

2.2 隐马尔科夫模型

HMM的三个假设：

对于一个随机事件，有一观察值序列： $O=O_1,O_2,\cdots O_T$

该事件隐含着一个状态序列： $Q = q_1,q_2,\cdots q_T$。

假设1： 马尔可夫性假设（状态构成一阶马尔可夫链）

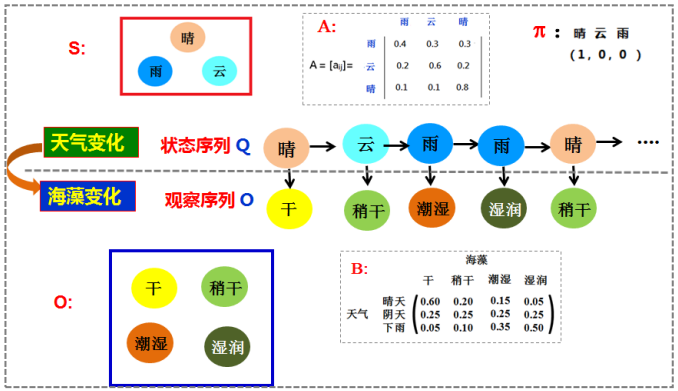
$$P(q_i|q_{i-1}\cdots q_1) = P(q_i|q_{i-1})$$

假设2： 不动性假设（状态与具体时间无关）

$$P(q_{i+1}|q_i) = P(q_{j+1}|q_j)，对任意i，j成立$$

假设3： 输出独立性假设（输出仅与当前状态有关）

$$p(O_1,\ldots,O_T \mid q_1,\ldots,q_T) = \Pi p(O_t \mid q_t)$$



2.2 隐马尔科夫模型

HMM五元组说明：

1. 隐藏状态s：一个系统的(真实)状态，可以由一个马尔科夫过程进行描述（如, 天气）

3. 观察状态 o：在这个过程中‘可视’的状态（例如，海藻的湿度）

3. 状态转移概率矩阵 $A = a_{ij}$ ：包含了一个隐藏状态到另一个隐藏状态的概率 。其中,

$$\begin{cases} a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \le i, j \le N \\ a_{ij} \ge 0 \\ \sum_{j=1}^{N} a_{ij} = 1 \end{cases}$$

4. 观察概率矩阵 $B=b_j(k)$: 从隐藏状态 S_j 观察到某一特定符号 v_k 的概率分布矩阵。

其中，

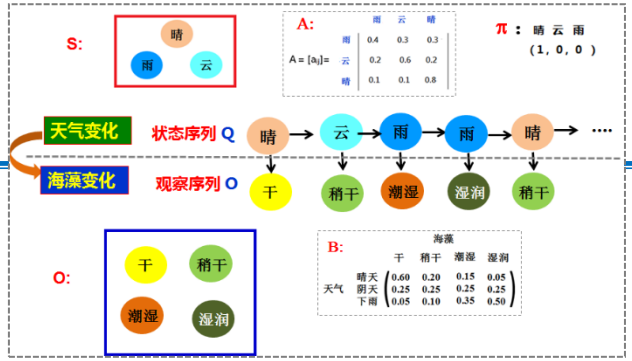
$$\begin{cases} b_j(k) = p(O_t = v_k \mid q_t = S_j), \quad 1 \le j \le N, \quad 1 \le k \le M \\ b_j(k) \ge 0 \\ \sum_{k=1}^{M} b_j(k) = 1 \end{cases}$$

2.2 隐马尔科夫模型

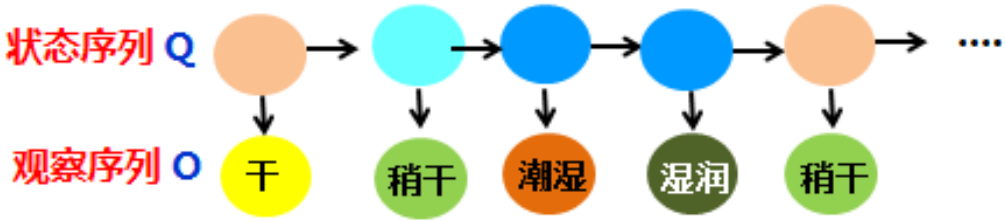
5. 初始状态的概率分布为：$\pi = \pi_i$，其中，

$$\begin{cases} \pi_i = p(q_1 = S_i), \quad 1 \le i \le N \\ \pi_i \ge 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases}$$

2.2 隐马尔科夫模型



隐马尔可夫模型结构(HMM):



输入： 观察序列

输出： 1. 观察序列的概率值 2. 隐状态序列

参数： $P(q_t|q_{t-1})$，$P(O_t|q_t)$

A矩阵　　B矩阵

函数关系：

(1) 观察序列的概率值 （ HMM评估问题 ）

(2) 隐状态序列 （ HMM解码问题 ）

（1）. HMM评估问题

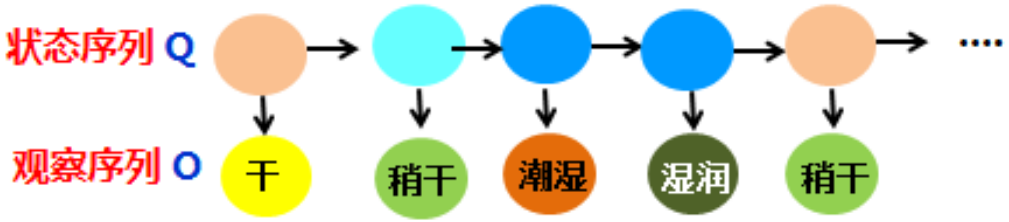
(1) HMM评估问题

HMM评估问题

对于给定观察序列 $O=O_1,O_2,\cdots O_T$，以及模型 $\lambda=(A,B,\pi)$

求观察序列的概率P(O|λ)

如：



求：观察序列概率P（O | λ）=?

问题：

1. 观察序列概率P(O|λ)定义
2. 如何计算 P(O|λ)

(1) HMM评估问题

1. 观察序列概率P(O|λ)定义

已知

S = {晴天，阴天 ，下雨 }

O = {湿润， 潮湿， 稍干，干燥 }

A

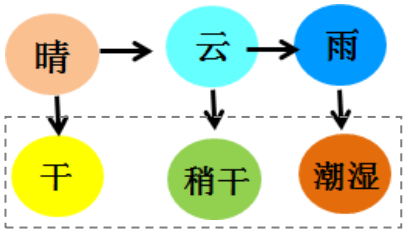
	晴天	阴天	下雨
晴天	0.50	0.25	0.25
阴天	0.375	0.25	0.375
下雨	0.25	0.125	0.625

B

天气	干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

$\pi = (1,\ 0,\ 0)$

- 对于给定的一个状态序列 $Q = q_1 q_2 \cdots q_T$，



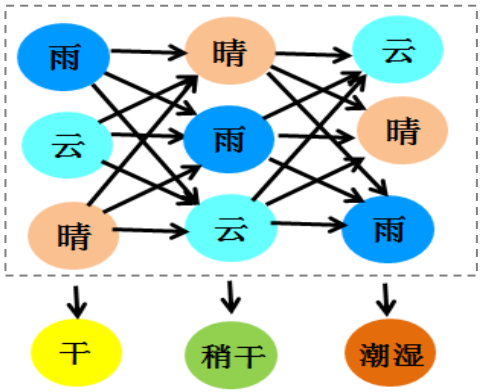
观察序列

$$P(O,Q \mid \lambda) = \pi_{q1} a_{q1q2} a_{q2q3} \cdots a_{qT-1qT} \; b_{q1}(O_1)\; b_{q2}(O_2) \cdots b_{qT}(O_T)$$

$P(Q \mid \lambda)$ $\quad$ $P(O \mid Q, \lambda)$

$$P(O,Q \mid \lambda) = P(Q \mid \lambda)\, P(O \mid Q, \lambda)$$

- 对于全部状态序列



观察序列来源

观察序列概率：

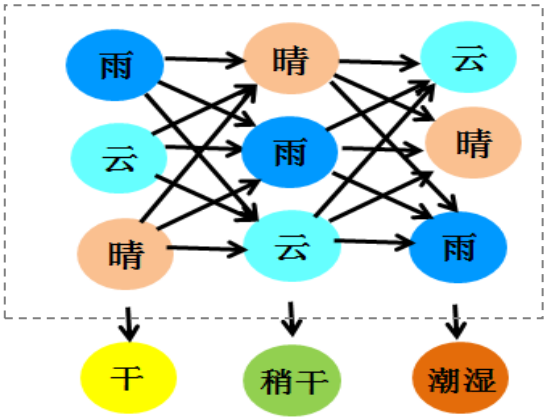
$$P(O \mid \lambda) = \sum_{Q} P(O,Q \mid \lambda) = \sum_{Q} P(Q \mid \lambda) P(O \mid Q, \lambda)$$

(1) HMM评估问题

2. 如何计算 P(O|λ)

观察序列概率：

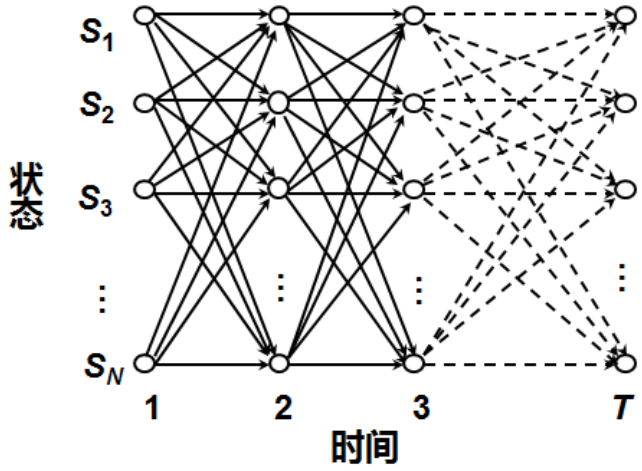
$$P(O \mid \lambda) = \sum_{Q} P(O, Q \mid \lambda) = \sum_{Q} P(Q \mid \lambda) P(O \mid Q, \lambda)$$



（1）穷举法：找到每一个可能的隐藏状态的序列，这里有3^3= 27种，可观察序列的概率就是这27种可能的和。 很显然，这种计算的效率非常低，尤其是当模型中的状态非常多或者序列很长的时候。

(1) HMM评估问题

穷举法的问题：



◆困难：

如果模型μ有 N 个不同的状态，时间长度为 T，那么有 N^T 个可能的状态序列，搜索路径成指数级组合爆炸

解决方法：

(2) 前向算法（后向算法）：利用动态规划使用递归来降低计算复杂度

(1) HMM评估问题

向前算法

(1) HMM评估问题

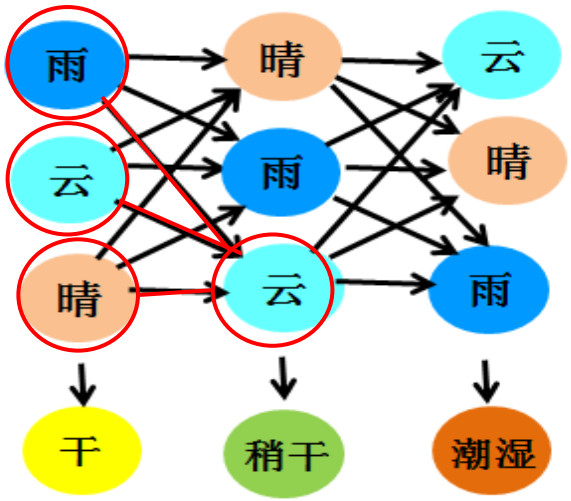
前向算法基本思想：

使用递归来降低计算复杂度

$$P(O \mid \lambda) = \sum_{Q} P(O, Q \mid \lambda) = \sum_{Q} P(Q \mid \lambda) P(O \mid Q, \lambda)$$

$t=1 \quad \pi_i$

$$\alpha_1(i) = \pi_i b_i(O_1)$$



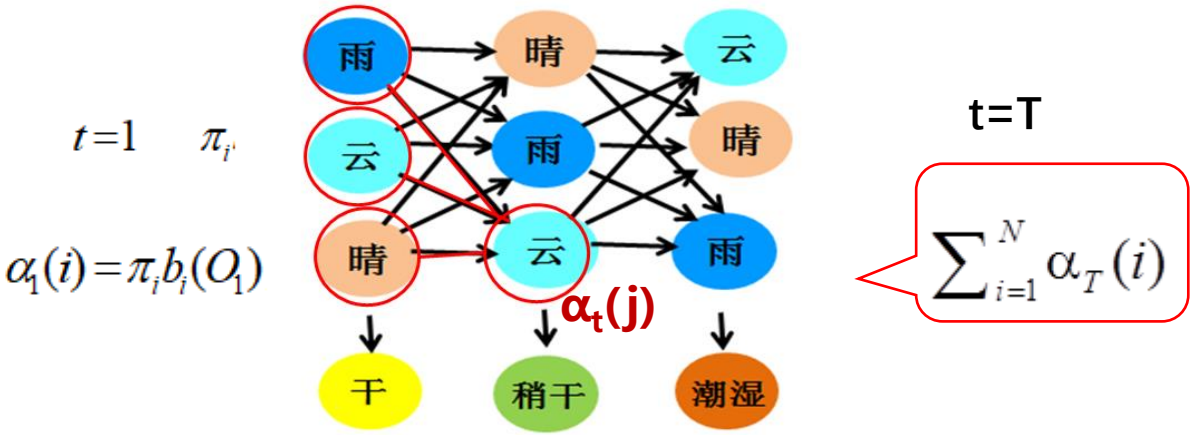
t=T

$$\sum_{i=1}^{N} \alpha_T(i)$$

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}] b_j(O_{t+1}),$$

（1） HMM评估问题

前向算法实现：



定义 前向变量 $\alpha_t(j)$（部分概率），表示达到某个中间状态的概率

- 当 t=1 时，是初始概率，Pr(状态 j | t=1) = π(状态 j)

$$\alpha_1(i)=\pi_i b_i(O_1),\quad t=1$$

- 当1 ≤ t ≤ T-1 时，

$$\alpha_t(j)=[\sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}]b_j(O_t),\quad 1\le t\le T-1$$

- 最终结果 $p(O\mid\mu)=\sum_{i=1}^{N}\alpha_T(i)$

(1) HMM评估问题

前向算法 **(The forward procedure)**

(1) 初始化：$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$

(2) 循环计算：

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1}), \qquad 1 \le t \le T-1$$

(3) 结束，输出：

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

(1) HMM评估问题

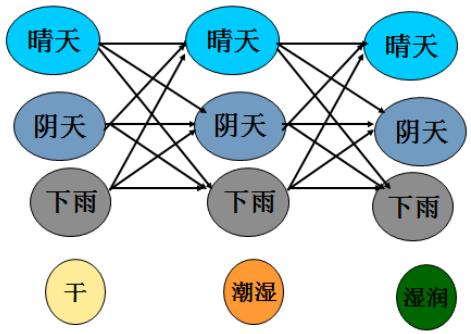
算法的时间复杂性：

每计算一个 $\alpha_t(i)$ 必须考虑从 t-1 时的所有 N 个状态转移到状态 S_i 的可能性，时间复杂性为 O(N)，对应每个时刻 t，要计算 N 个前向变量：$\alpha_t(1)$, $\alpha_t(2)$, …, $\alpha_t(N)$，所以，时间复杂性为：$O(N)\times N = O(N^2)$。 又因 t = 1, 2, …, T，所以前向算法总的复杂性为：$O(N^2T)$

穷举算法的时间开销是和 T 指数相关　即 $O(N^T)$

(1) HMM评估问题

例1： 已有天气和海藻关系的HMM模型 λ ；求连续3 天海藻湿度的观察结果是（干燥、潮湿、湿润）的概率。



S = {晴天，阴天 ，下雨 }

O = {湿润， 潮湿， 稍干，干燥 }

A

	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

B

天气 \ 海藻	干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

π （1，0，0）

(1) HMM评估问题

解： 用向前算法

	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

天气

海藻	干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

$\pi = (1, 0, 0)$

1. 前向算法

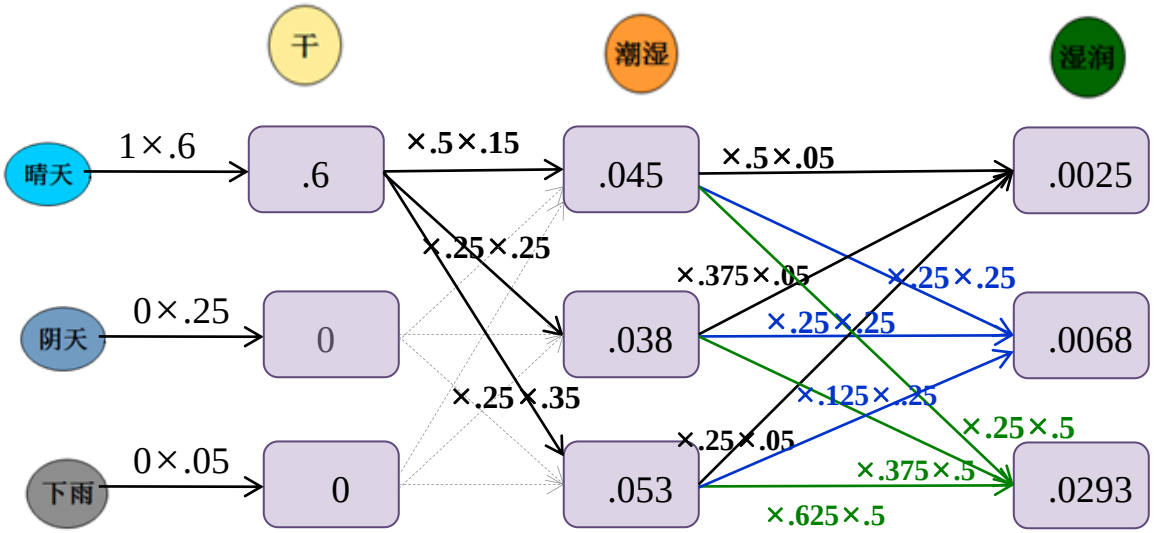
(1) 初始化： $\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$

(2) 循环计算：

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1}), \quad 1 \le t \le T-1$$

(3) 结束，输出：

$$p(O|\mu) = \sum_{i=1}^{N} \alpha_T(i)$$



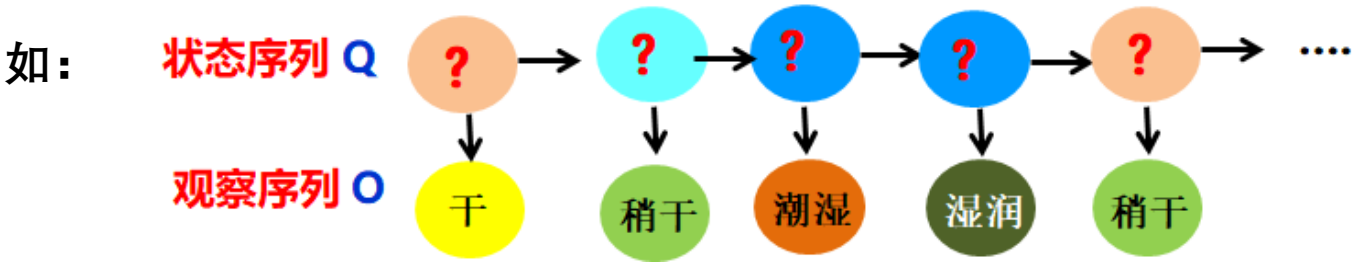
$$p(O|\mu) = \sum_{i=1}^{N} \alpha_T(i) \quad = 0.0025 + 0.0068 + 0.0293 \quad = 0.0386$$

(2) HMM解码问题

(2) HMM解码问题

HMM解码问题:

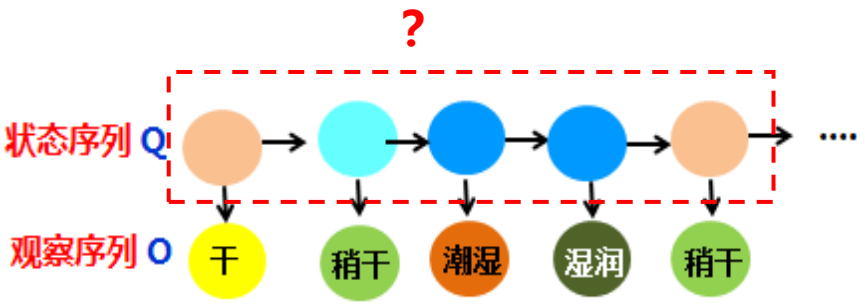
对于给定观察序列 $O=O_1,O_2,\cdots O_T$，以及模型 $\lambda=(A,B,\pi)$ 如何选择一个对应的状态序列 $S=S_1,S_2,\cdots S_T$，使得S能够最为合理的解释观察序列 O

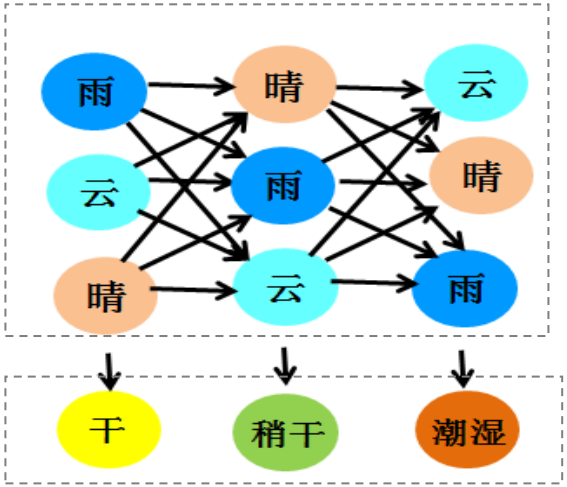


求：状态序列序列 $S=S_1,S_2,\cdots S_T$

(2) HMM解码问题

求：状态序列序列 $S = S_1, S_2, \ldots S_T$





（1）穷举法：找到每一个可能产生观察序列的状态序列，这里有3^3= 27种，计算每种可能情况下观察序列的概率，概率最大的状态序列就是要找的状态序列。 很显然，这种计算的效率非常低，尤其是当模型中的状态非常多或者序列很长的时候。

解决方法：

（2）Viterbi 搜索算法：利用动态规划使用递归来降低计算复杂度

(2) HMM解码问题

Viterbi 搜索算法

Andrew Viterbi

安德鲁.维特比（Andrew Viterbi ）

1967年发明了维特比算法。

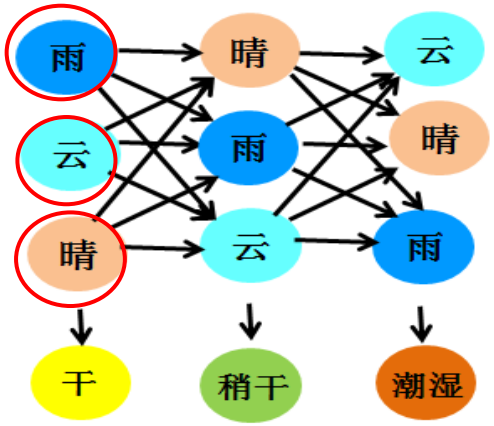
维特比算法： 利用动态规划方法解决特殊的篱笆网络有向图的最短路径问题。 是现代数字通信中使用最频繁的算法；同时也是很多自然语言处理的解码算法。

(2) HMM解码问题

Viterbi 算法基本思想： 使用递归来降低复杂度

$t=1$ π_i

$\alpha_1(i)=\pi_i b_i(O_1)$



t=T

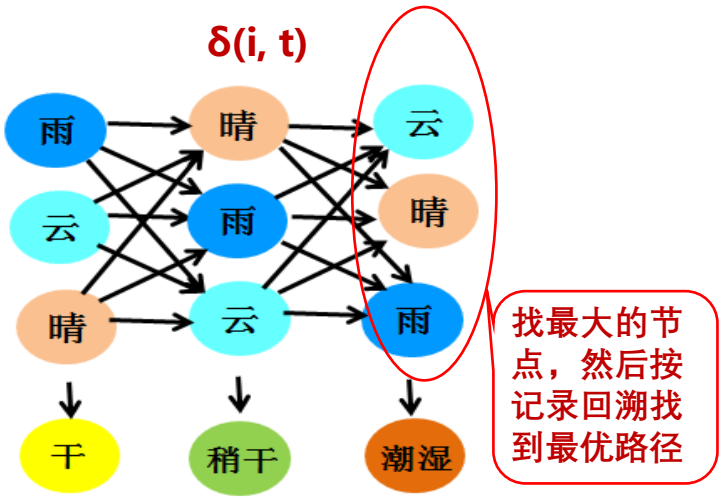
取概率最大的点进行回溯得到最大序列

1. 如果概率最大路径（或说最短路径）经 i 时刻某个点，一定可以找到S到该点的最短路径（可将i时刻点的最短路径记录）
2. 从S到E 的路径必定经过 i时刻的某个点
3. 当从状态 i 进入到i+1状态时计算S到i+1 状态时，只考虑 i状态所有节点最短路径和和它们到 i+1状态的距离即可。

Viterbi时间复杂度： $O(N^2T)$ **穷举法：** $O(N^T)$

(2) HMM解码问题

Viterbi 算法实现：



(1) 部分最优路径概率

定义一个部分概率δ ;用 δ(i, t) 来表示在t时刻，到状态i的所有可能的序列（路径）中概率最大的序列的概率

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N$$

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(O_t), \quad 2 \le t \le T, \quad 1 \le j \le N$$

(2)向后指针记录最优路径

利用一个后向指针 φ 来记录导致某个状态最大局部概率的前一个状态

$$\phi_t(i) = \arg\max_j(\delta_{t-1}(j)a_{ji})$$

(3) 结果

$$q_t^* = \varphi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1$$

(2) HMM解码问题

Viterbi 搜索算法

1. 初始化： $\delta_1(i) = \pi_i b_i(O_1)$， $\varphi_1(i) = 0$， $1 \le i \le N$

2. 递归： $\delta_t(j) = [\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij}] b_j(O_t)$， $2 \le t \le T, 1 \le j \le N$

   $\varphi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$， $2 \le t \le T, 1 \le j \le N$

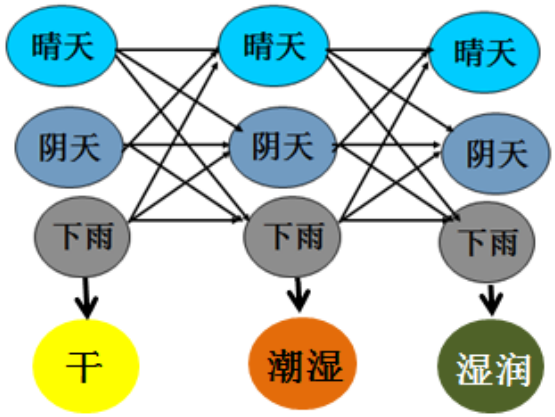
3. 终结： $p^* = \max_{1 \le i \le N} [\delta_T(i)])$ ， $q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$

4. 路径回溯： $q_t^* = \varphi_{t+1}(q_{t+1}^*)$， $t = T-1, T-2, \ldots, 1$

算法的时间复杂度： $O(N^2 T)$

(2) HMM解码问题

例1： 已有天气和海藻关系的HMM模型 λ 和连续3 天海藻湿度的观察结果（干燥、潮湿、湿润），求最可能的天气序列



A：

	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

B：

		海藻			
		干	稍干	潮湿	湿润
天气	晴天	0.60	0.20	0.15	0.05
	阴天	0.25	0.25	0.25	0.25
	下雨	0.05	0.10	0.35	0.50

设： $\pi=(1, 0, 0)$

(2) HMM解码问题

解：用Viterbi 搜索算法

	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

海藻

天气	干	稍干	潮湿	湿润
晴天	0.60	0.20	0.15	0.05
阴天	0.25	0.25	0.25	0.25
下雨	0.05	0.10	0.35	0.50

$\pi = (1, 0, 0)$

Viterbi 搜索算法

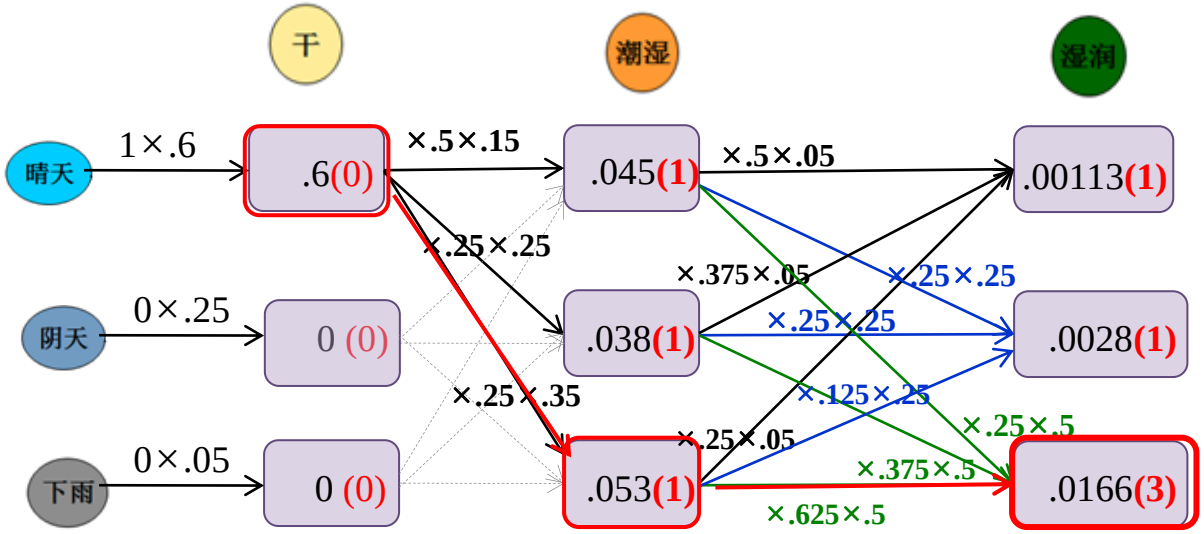
1. 初始化： $\delta_1(i) = \pi_i b_i(O_1)$， $\varphi_1(i) = 0$， $1 \le i \le N$

2. 递归： $\delta_t(j) = [\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij}] b_j(O_t)$， $2 \le t \le T, 1 \le j \le N$

$$\varphi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \le t \le T, 1 \le j \le N$$

3. 终结： $p^* = \max_{1 \le i \le N} [\delta_T(i)]$， $q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$

4. 路径回溯： $q_t^* = \varphi_{t+1}(q_{t+1}^*)$， $t = T-1, T-2, \ldots, 1$



（干燥、潮湿、湿润），最可能的天气序列：（晴、雨、雨）

2.2 隐马尔科夫模型

HMM参数学习

隐马尔科夫模型参数

$$P(S_t \mid S_{t-1}) = \frac{P(S_{t-1} S_t)}{P(S_{t-1})} \qquad P(O_t \mid S_t) = \frac{P(O_t, S_t)}{P(S_t)}$$

训练思路：

通过观察序列 O = $O_1O_2 \cdots O_T$作为训练数据，用最大似然估计，使得观察序列 O 的概率 p(O|μ) 最大。

2.2 隐马尔科夫模型

情况1： 产生观察序列 O 的状态 $Q = q_1 q_2 \cdots q_T$ 已知，可以采用**有监督**的学习方法，用**最大似然估计**来计算 μ 的参数：

$$\overline{\pi}_i = \delta(q_1, S_i)$$

$$\overline{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的次数}}{Q\text{中所有从状态}q_i\text{转移到另一状态(包括}q_j\text{自身)的总数}} = \frac{\sum_{t=1}^{T-1}\delta(q_t, S_i)\times\delta(q_{t+1}, S_j)}{\sum_{t=1}^{T-1}\delta(q_t, S_i)}$$

其中，δ(x, y) 为克罗奈克(Kronecker)函数，当 x= y 时, δ(x, y)=1, 否则δ(x, y) = 0。

$$\overline{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的次数}}{Q\text{到达}q_j\text{的总次数}} = \frac{\sum_{t=1}^{T}\delta(q_t, S_j)\times\delta(O_t, v_k)}{\sum_{t=1}^{T}\delta(q_t, S_j)}$$

其中，v_k 是模型输出符号集中的第 k 个符号。

2.2 隐马尔科夫模型

情况2：HMM 中的状态序列Q 是观察不到的，这时，最大似然估计方法不可行。可以采用无监督的EM学习方法。

解决方法：

期望最大化EM 算法。根据EM 算法调节模型的参数 π_i a_{ij} ，$b_{j(k)}$，使得观察序列O的概率P(O|M)最大，主要使用前向后向算法（鲍姆-韦尔奇Baum-Welch ）算法。

(略)

2.2 隐马尔科夫模型

统计自然语言处理时代HMM模型在统计自然语言处理中有着广泛的应用

观察序列$O=O_1O_2\cdots O_T$： 处理的语言单位，一般为 词

状态序列 S =S1S2…ST： 与语言单位对应的句法信息，一般为 词类/词性

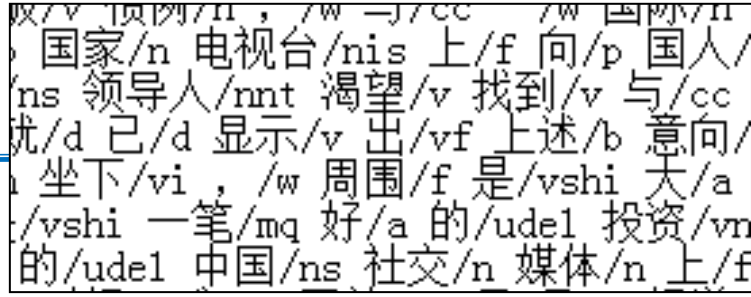
模型参数：初始状态概率、状态转移概率、发射概率 需要学习获得

★ 分词： HMM的评估问题：当分词出现多种可能时，求观察序列 $O=O_1O_2\cdots O_T$ 的概率， 结果取 概率最大的序列 ；解码问题：用序列标注直接进行分词

★ 词性标注： 相当HMM的解码问题。即求观察序列 $O=O_1O_2\cdots O_T$ 下，概率最大的标注序列 argmax P(Q|O ,μ)

★ 其他：如 短语识别、语音识别 …….

2.2 隐马尔科夫模型



例1： HMM模型在词性标注中的应用

设，有如下从语料库训练得到的词性转移概率矩阵和词语生成概率矩阵

词性转移概率

词性	估计
PROB(ART | ø)	0.71
PROB(N | ø)	0.29
PROB(N | ART)	1
PROB(V | N)	0.43
PROB(N | N)	0.13
PROB(P | N)	0.44
PROB(N | V)	0.35
PROB(ART | V)	0.65
PROB(ART | P)	0.74
PROB(N | P)	0.26

词语生成概率

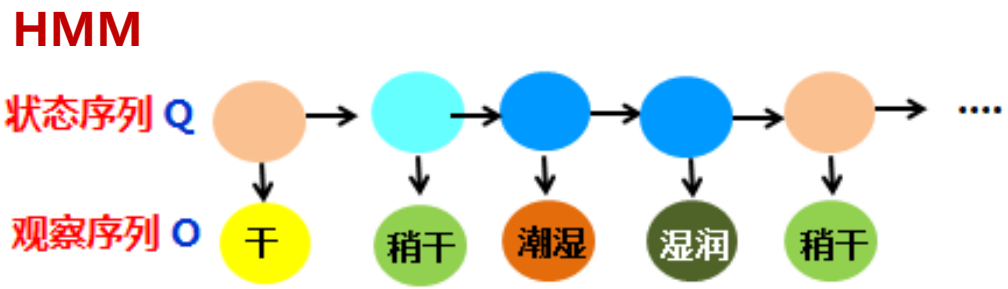
PROB(*the* | ART)	0.54
PROB(*flies* | N)	0.025
PROB(*flies* | V)	0.076
PROB(*like* | V)	0.1
PROB(*like* | P)	0.068
PROB(*like* | N)	0.012
PROB(*a* | ART)	0.360
PROB(*a* | N)	0.001
PROB(*flower* | N)	0.063
PROB(*flower* | V)	0.05
PROB(*birds* | N)	0.076

试对 “ flies like a flower “ 进行词性标注

2.2 隐马尔科夫模型

解： **问题求解目标**：对每个词标出其词性

该问题属于序列标注问题，可用HMM模型进行标注



观察集（词集）： flies， like， a， flower

状态集（词性集）： N，V，P，ART

可用 Viterbi 搜索算法 解码

词性转移概率

词性	估计
PROB(ART | ø)	0.71
PROB(N | ø)	0.29
PROB(N | ART)	1
PROB(V | N)	0.43
PROB(N | N)	0.13
PROB(P | N)	0.44
PROB(N | V)	0.35
PROB(ART | V)	0.65
PROB(ART | P)	0.74
PROB(N | P)	0.26

词语生成概率

PROB(*the* | ART)	0.54
PROB(*flies* | N)	0.025
PROB(*flies* | V)	0.076
PROB(*like* | V)	0.1
PROB(*like* | P)	0.068
PROB(*like* | N)	0.012
PROB(*a* | ART)	0.360
PROB(*a* | N)	0.001
PROB(*flower* | N)	0.063
PROB(*flower* | V)	0.05
PROB(*birds* | N)	0.076

Viterbi 搜索算法

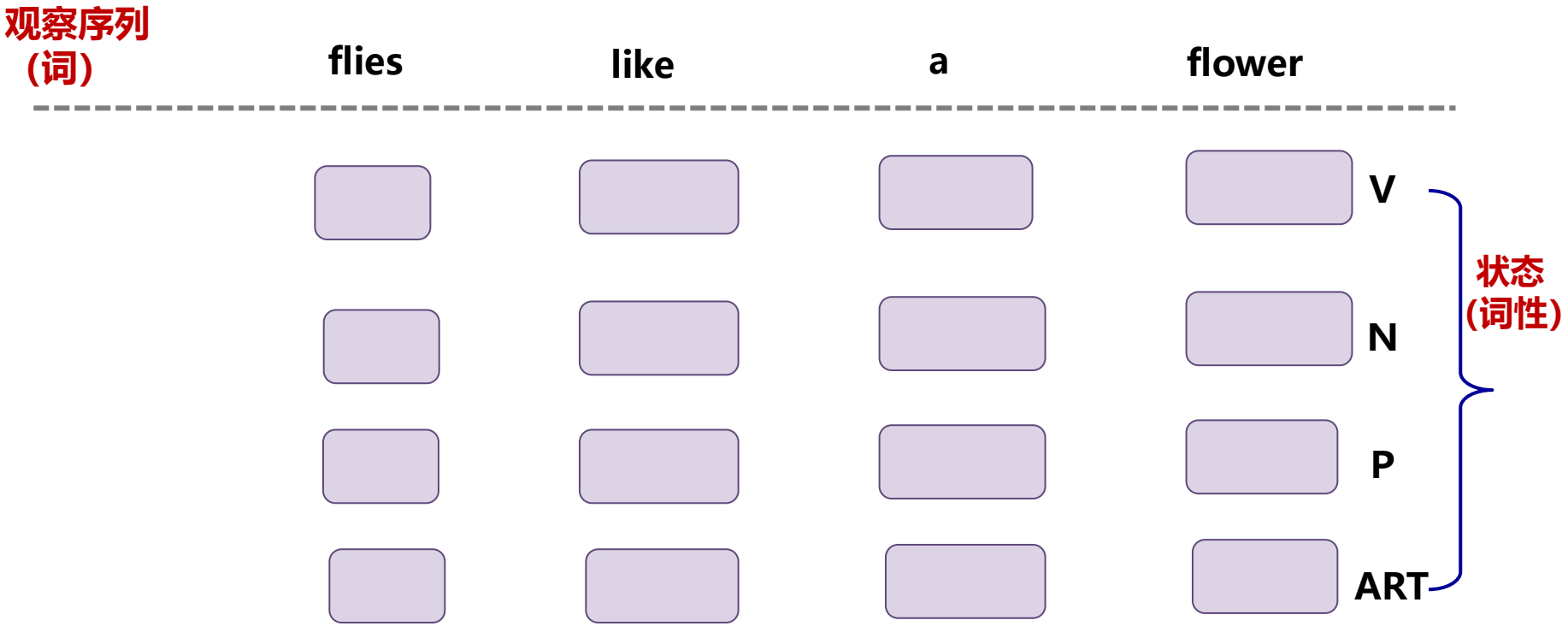
1. 初始化： $\delta_1(i)=\pi_i b_i(O_1)$， $\varphi_1(i)=0$， $1\le i\le N$

2. 递归： $\delta_t(j)=[\max_{1\le i\le N}\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

   $\varphi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

3. 终结： $p^*=\max_{1\le i\le N}[\delta_T(i)]$ ， $q_T^*=\arg\max_{1\le i\le N}[\delta_T(i)]$

4. 路径回溯： $q_t^*=\varphi_{t+1}(q_{t+1}^*)$， $t=T-1,T-2,\ldots,1$



词性转移概率

词性	估计
PROB(ART | ø)	0.71
PROB(N | ø)	0.29
PROB(N | ART)	1
PROB(V | N)	0.43
PROB(N | N)	0.13
PROB(P | N)	0.44
PROB(N | V)	0.35
PROB(ART | V)	0.65
PROB(ART | P)	0.74
PROB(N | P)	0.26

词语生成概率

PROB(*the* | ART)	0.54
PROB(*flies* | N)	0.025
PROB(*flies* | V)	0.076
PROB(*like* | V)	0.1
PROB(*like* | P)	0.068
PROB(*like* | N)	0.012
PROB(*a* | ART)	0.360
PROB(*a* | N)	0.001
PROB(*flower* | N)	0.063
PROB(*flower* | V)	0.05
PROB(*birds* | N)	0.076

Viterbi 搜索算法

1. 初始化： $\delta_1(i)=\pi_i b_i(O_1)$， $\varphi_1(i)=0$， $1\le i\le N$

2. 递归： $\delta_t(j)=[\max_{1\le i\le N}\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

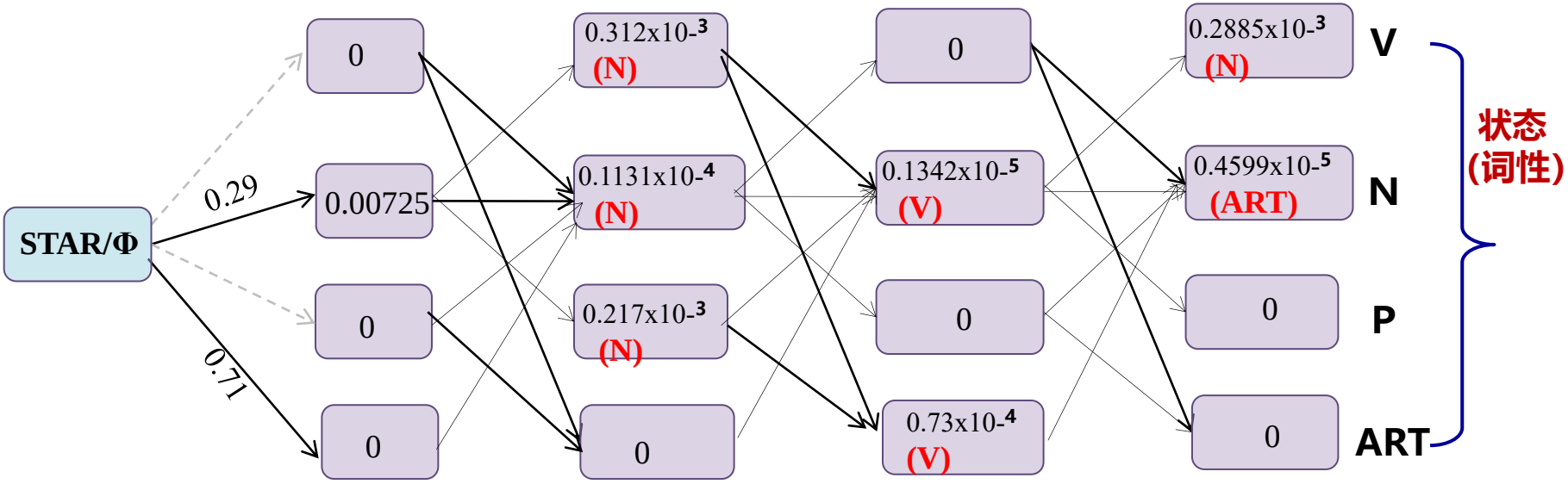
   $\varphi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

3. 终结： $p^*=\max_{1\le i\le N}[\delta_T(i)]$ ， $q_T^*=\arg\max_{1\le i\le N}[\delta_T(i)]$

4. 路径回溯： $q_t^*=\varphi_{t+1}(q_{t+1}^*)$， $t=T-1,T-2,\ldots,1$

观察序列（词）

flies　　like　　a　　flower



词性转移概率

词性	估计
PROB(ART | ø)	0.71
PROB(N | ø)	0.29
PROB(N | ART)	1
PROB(V | N)	0.43
PROB(N | N)	0.13
PROB(P | N)	0.44
PROB(N | V)	0.35
PROB(ART | V)	0.65
PROB(ART | P)	0.74
PROB(N | P)	0.26

词语生成概率

PROB(*the* | ART)	0.54
PROB(*flies* | N)	0.025
PROB(*flies* | V)	0.076
PROB(*like* | V)	0.1
PROB(*like* | P)	0.068
PROB(*like* | N)	0.012
PROB(*a* | ART)	0.360
PROB(*a* | N)	0.001
PROB(*flower* | N)	0.063
PROB(*flower* | V)	0.05
PROB(*birds* | N)	0.076

Viterbi 搜索算法

1. 初始化： $\delta_1(i)=\pi_i b_i(O_1)$， $\varphi_1(i)=0$， $1\le i\le N$

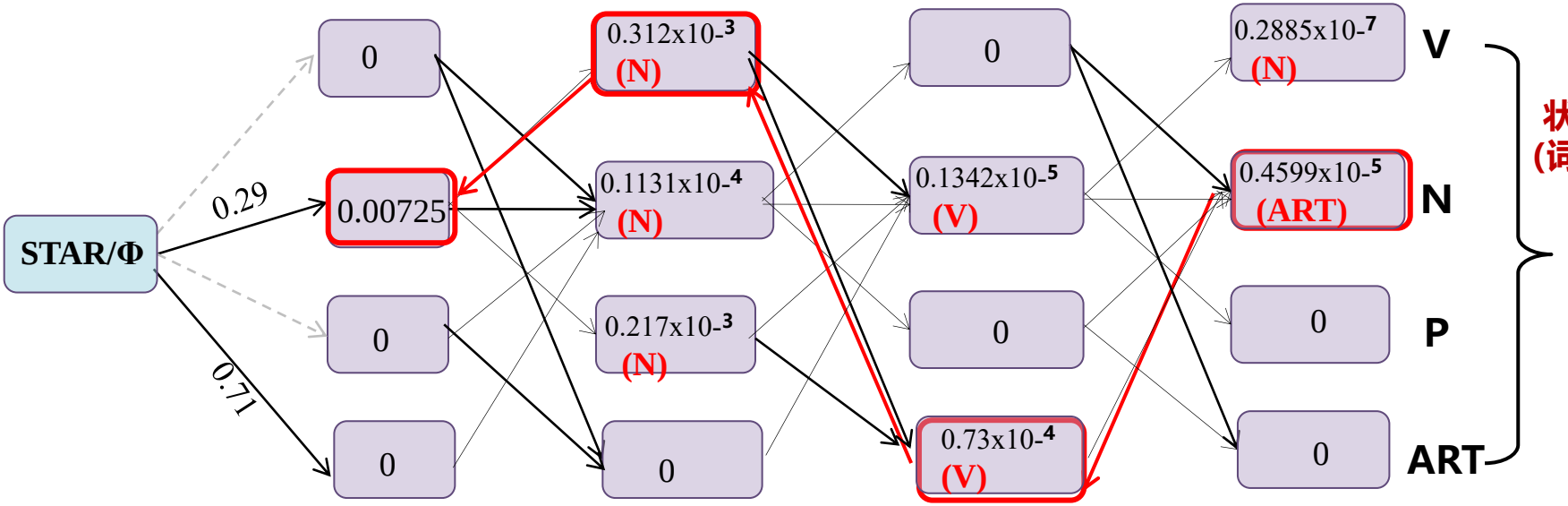
2. 递归： $\delta_t(j)=[\max_{1\le i\le N}\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

   $\varphi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t)$， $2\le t\le T, 1\le j\le N$

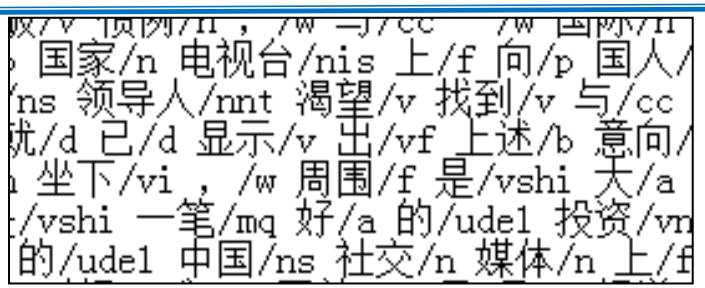
3. 终结： $p^*=\max_{1\le i\le N}[\delta_T(i)]$ ， $q_T^*=\arg\max_{1\le i\le N}[\delta_T(i)]$

4. 路径回溯： $q_t^*=\varphi_{t+1}(q_{t+1}^*)$， $t=T-1,T-2,\ldots,1$

观察序列（词）：flies /N　like /V　a /ART　flower /N　结果



2.2 隐马尔科夫模型



语料

例2： 用HMM实现简单的中文分词

例 . 输入： 北京是中国的首都

输出：北京 是 中国 的 首都 (词序列)

解 ： 用单字序列标注方法

{ 词首/B，词内/I，词尾/E，单字词/O }

模型HMM:

S：状态集合， { B，I，E，O}

O：观察值集合，{单个汉字： 人、民、中.....}

A：状态转移概率矩阵

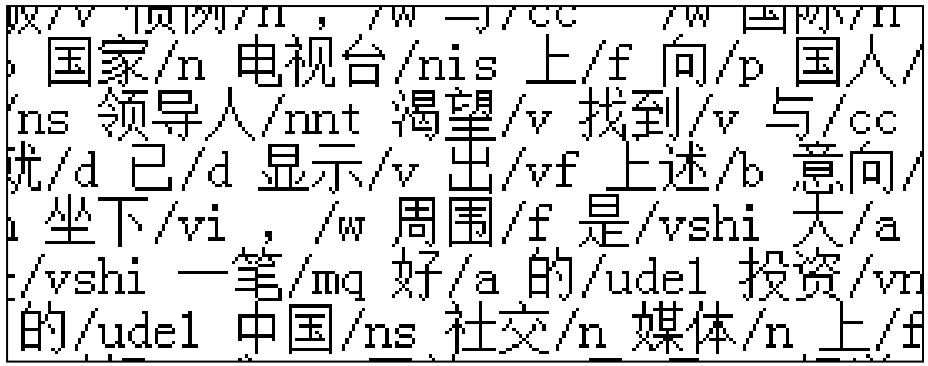
B：给定状态下，观察值的概率分布

π：初始状态空间的概率分布

2.2 隐马尔科夫模型

参数学习

语料：



训练语料：

国/B 家/E 电/B 视/I 台/E 上/O 向/O 国/B人/E 领/B 导/I 人/E....

2.2 隐马尔科夫模型

训练语料：

国/B 家/E 电/B 视/I 台/E 上/O 向/O
国/B人/E 领/B 导/I 人/E....

假设，语料中不重复的中文单字共8000个

- $A =$

	B	I	E	O
B	0	0.3	0.7	0
I	0	0.4	0.6	0
E	0.4	0	0	0.6
O	0.5	0	0	0.5

$A \in R^{4\times4}$，每行元素之和为1

- $B =$

	国	家	电	视	台	上	向	国	人	....
B	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	
I	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	....
E	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	....
O	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	....

$B \in R^{4\times8000}$，每行元素之和为1

- $\pi = [XXX, 0, 0, XXX]^T$ $\pi \in R^4$，元素之和为1

2.2 隐马尔科夫模型

用最大似然估计学习参数：

有观察序列$O=O_1O_2...O_T$ 和 状态序列$Q=q_1q_2.....q_T$

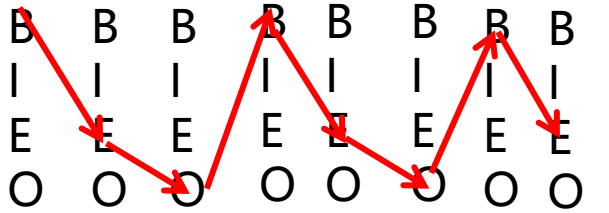
用极大似然估计

- $\pi_i = \frac{\sum_{t=1}^{T} \delta(q_t, S_i)}{T}$, $(S_0=\mathrm{B}, S_1=\mathrm{I}, S_2=\mathrm{E}, S_3=\mathrm{O})$
- $a_{ij} = \frac{\sum_{t=1}^{T-1} \delta(q_t, S_i) \times \delta(q_{t+1}, S_j)}{\sum_{t=1}^{T-1} \delta(q_t, S_i)}$
- $b_{jk} = \frac{\sum_{t=1}^{T} \delta(q_t, S_j) \times \delta(o_t, v_k)}{\sum_{t=1}^{T} \delta(q_t, S_j)}$

2.2 隐马尔科夫模型

预测-分词

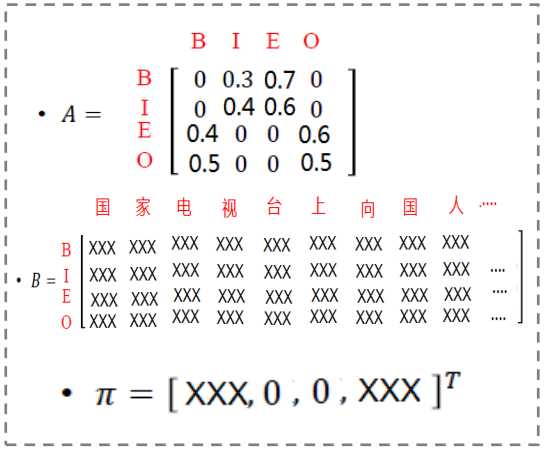
Viterbi算法



输入： 北 京 是 中 国 的 首 都

输出： B E O B E O B E

分词结果： 北京/ 是/ 中国/ 的/ 首都



注意： 分词和词性标注虽均用HMM模型， 但 状态集 观察集 不同，训练语料标注不同，模型参数不同

2.2 隐马尔科夫模型

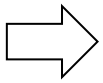
思考：

用HMM模型是否可以同时识别人名，地名和组织结构名？

答： 可以

在标签集中加入类别

{ B I O}



B-PER
I-PER
B-org
I-org
B-Loc
I-Loc
O

张/B-PER 三/I-PER 在/O 北/B-Loc 京/I-Loc 旅/O 游/O …..

隐马尔科夫模型问题

HMM 等生产式模型存在的问题

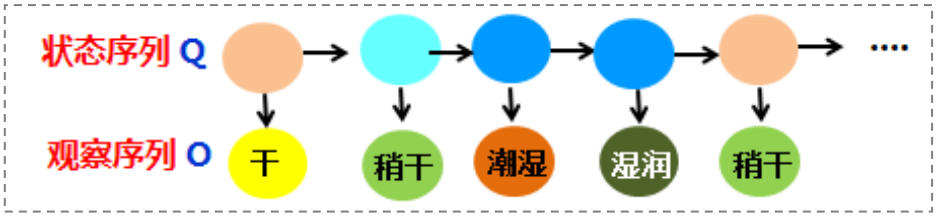
1. 由于生成模型定义的是联合概率，必须列举所有观察序列的可能值，这对多数领域来说是比较困难的。

在自然语言处理中，常知道各种各样但又不完全确定的信息，需要一个统一的模型将这些信息综合起来。

2. 输出独立性假设要求序列数据严格相互独立才能保证推导的正确性，导致其不能考虑上下文特征

在自然语言处理中，常常需要考虑上下文关系。

HMM

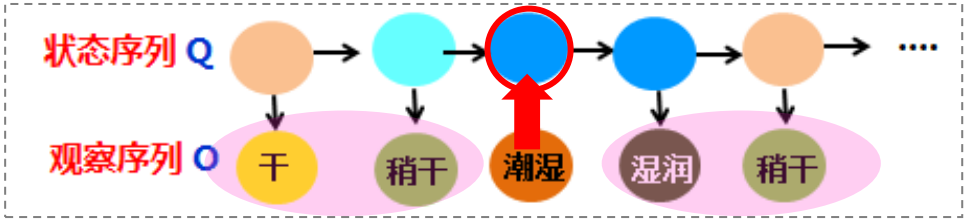


问题：HMM对于自然语言中的上下文信息不能利用

隐马尔科夫模型改进

最大熵模型：

1： 如何利用各种各样但又不完全确定的信息（上下文信息）？



如何建模上下文信息?

最大熵模型

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

其中:

$$Z_\lambda(x) = \sum_y \exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

称为归一化因子。

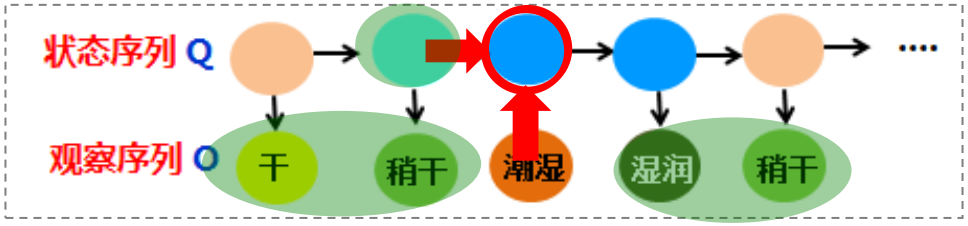
特点： 可以综合上下文信息

详情略

隐马尔科夫模型改进

条件随机场 CREF：

2： 如何用这些信息进行序列标注（生成序列上下文有关）



如何建模输出间信息？

条件随机场

$$p(y|x,\lambda)=\frac{1}{Z(x)}\exp\left(\sum_{ji}\lambda_j t_j(y_{i-1},y_i,x,i)+\sum_{ki}\mu_k s_k(y_i,x,i)\right)$$

其中：$$Z(x)=\sum_{y}\exp\left(\sum_{ji}\lambda_j t_j(y_{i-1},y_i,x,i)+\sum_{ki}\mu_k s_k(y_i,x,i)\right)$$

特点： 综合上下文信息

并且建立输出之间联系

详情略

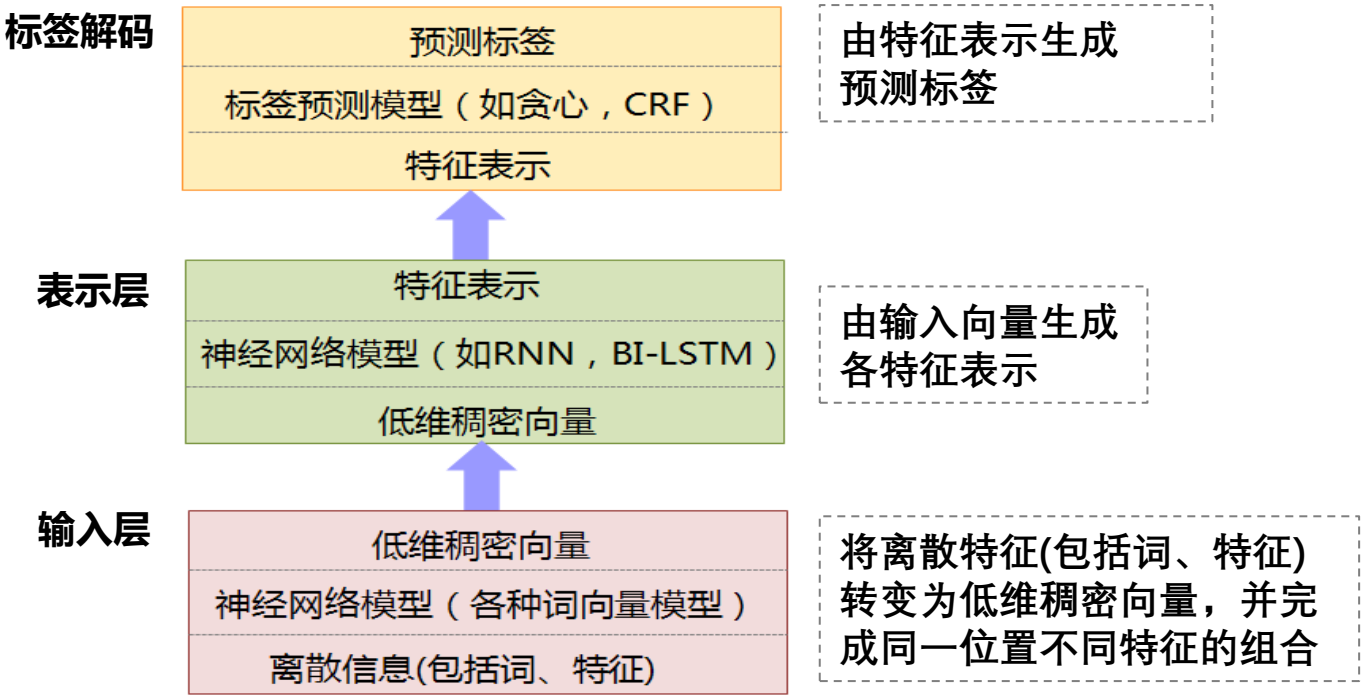
■ 序列标注

本节内容：

1. 序列标注问题概述
2. 隐马尔科夫模型HHM(概率统计模型)
3. 神经网络序列标注模型（深度学习模型）
   （1）双向RNN+softmax 模型
   （2）双向RNN+CRF 模型

(1) 双向RNN+softmax 模型：

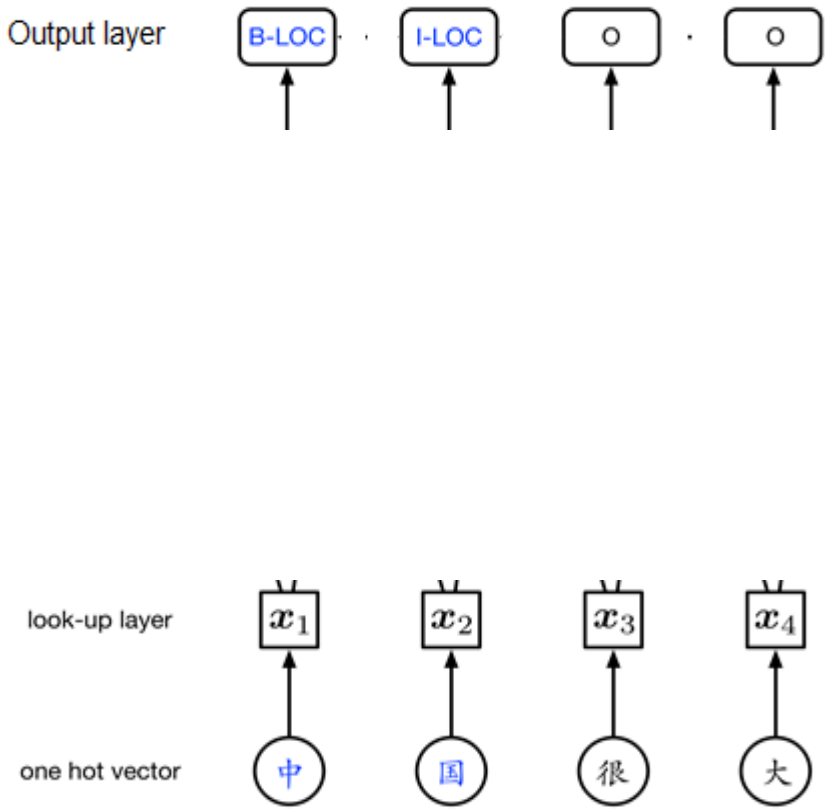
神经网络序列标注模型架构



(1) 双向RNN+softmax 模型：

(1) BiRNN+softmax 模型：

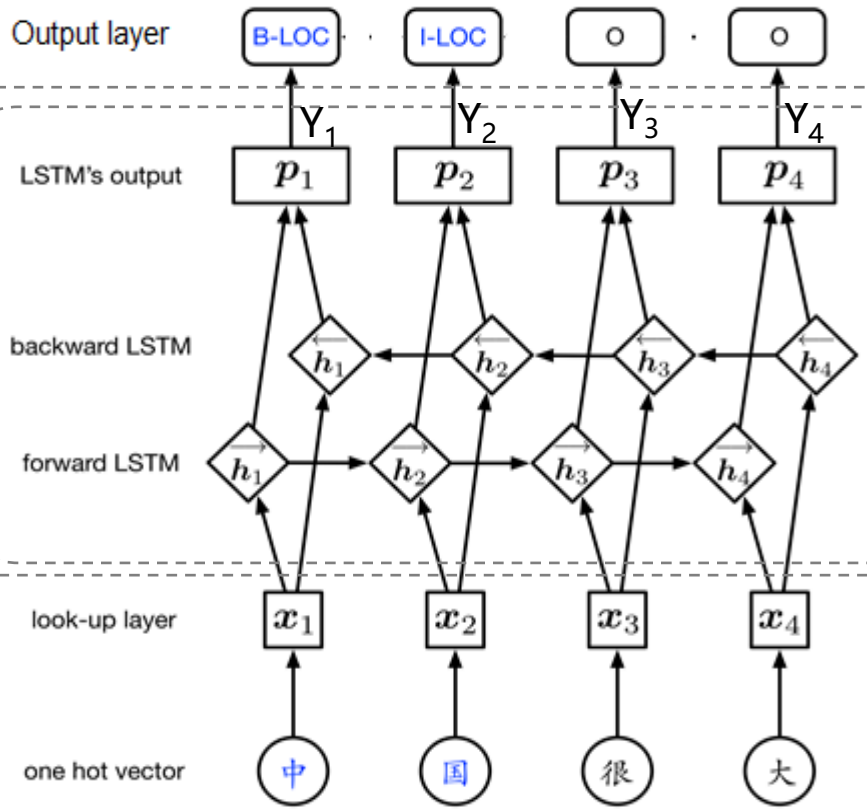
- **模型结构：**



(1) 双向RNN+softmax 模型：

(1) BiRNN+softmax 模型：

■ **模型结构：**



输出层： $Out_i = Max(y_i)$ 标签序列

$Y_i = softmax(p_i)$ (标签分布)

$p_i = \tanh(W_C[F(h_i), B(h_i)])$

反向隐藏层：

$$B(h_i) = \sigma((W2 \bullet X_i) + U2 \bullet B(h_{i+1}))$$

正向隐藏层：

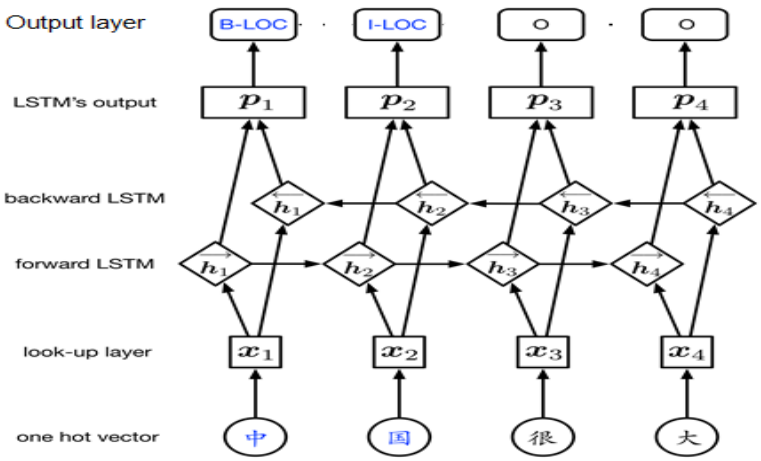
$$F(h_i) = \sigma((W1 \bullet X_i) + U1 \bullet F(h_{i-1}))$$

输入层： X_i ： 字 w_i 的字向量序列

参数： W1,U1, W2, U2 ,Wc,

(1) 双向RNN+softmax 模型：

■ 模型学习 (有监督)



$\hat{Y}$ 格式：(10000) (01000) (00000) (00100)

$\hat{Y}$: B-PER I-PER O B-Loc

X: 张 三 在 北 京

如 标人名： 训练数据（有标注训练集）

张/B-PER 三/I-PER 在/O 北/B-Loc 京/I-Loc 旅/O 游/O …..

标签集：{B-PER， I-PER， B-Loc， I-Loc， O}

(1) 双向RNN+softmax 模型：

■ 模型学习 (有监督)

- 定义损失函数

交叉熵损失：$J(\theta;x,y)=-\sum_{j=1}^{k} y_j \log((y_{pred})_j)$　　k 标签数

整体损失：$J(\theta)=\frac{1}{m}\sum_{i=1}^{m} J(\theta;x^{(i)},y^{(i)})$　　$\theta=\{W1,U1,W2,U2,Wc,\}$

- 用BPTT算法训练参数 W1,U1, W2, U2, Wc

■ 序列标注

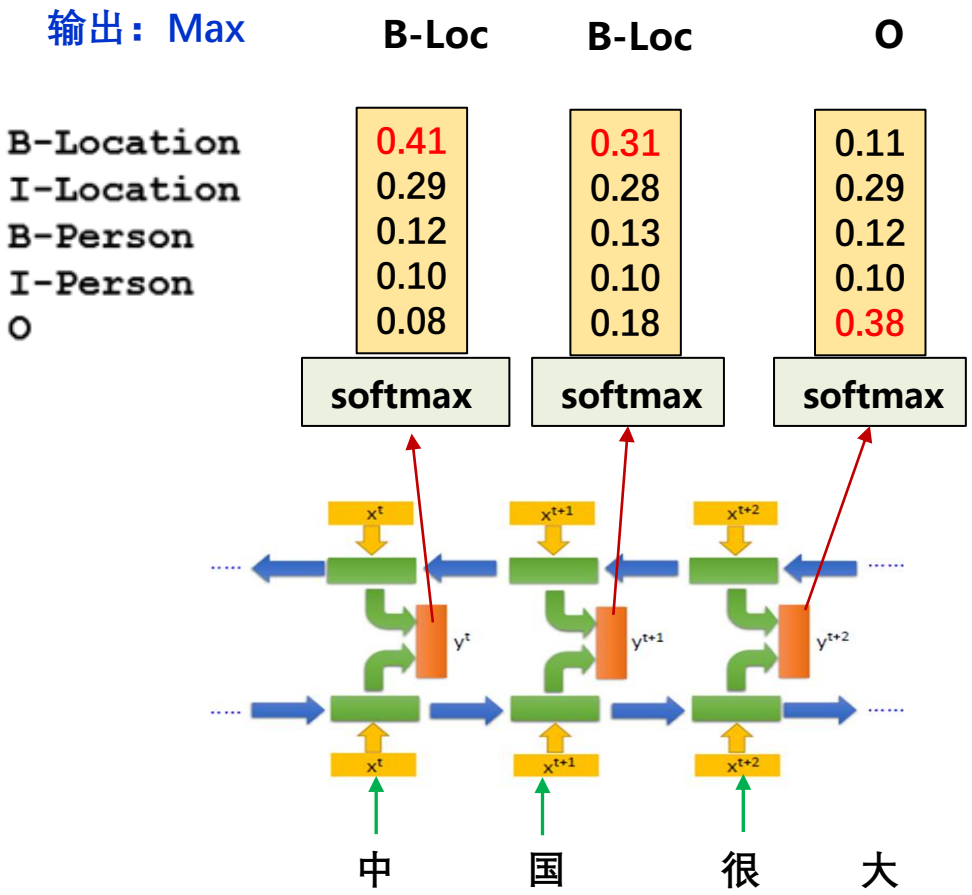
本节内容：

1. 序列标注问题概述
2. 隐马尔科夫模型HHM(概率统计模型)
3. 神经网络序列标注模型（深度学习模型）
   （1）双向RNN+softmax 模型
   （2）双向RNN+CRF 模型

(2) 双向RNN+CRF 模型

BiRNN+softmax 模型存在问题：

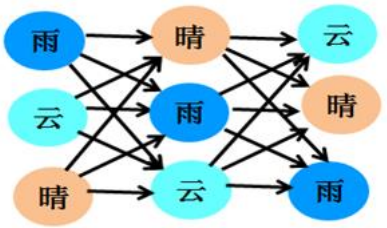
例如：



原因：输出独立

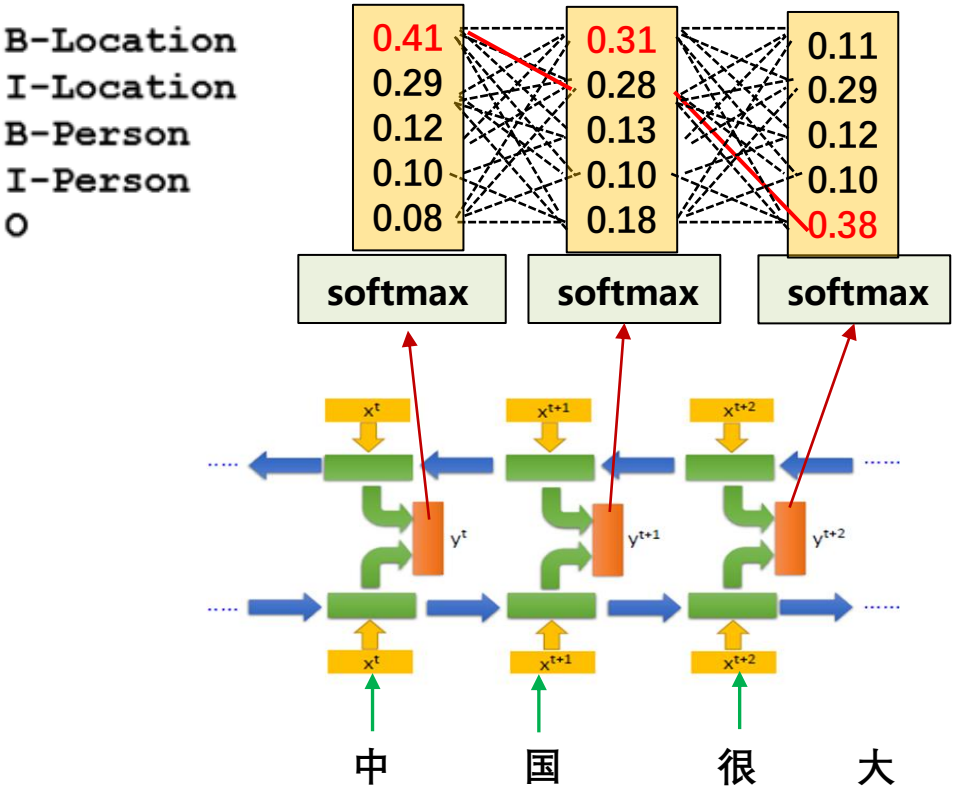
(2) 双向RNN+CRF 模型

对BiRNN+softmax 模型改进：



改进思路： 建立输出之间的关系

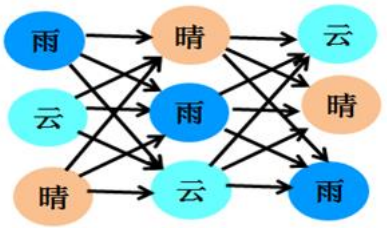
输出：概率最大的序列　B-Loc　I-Loc　O



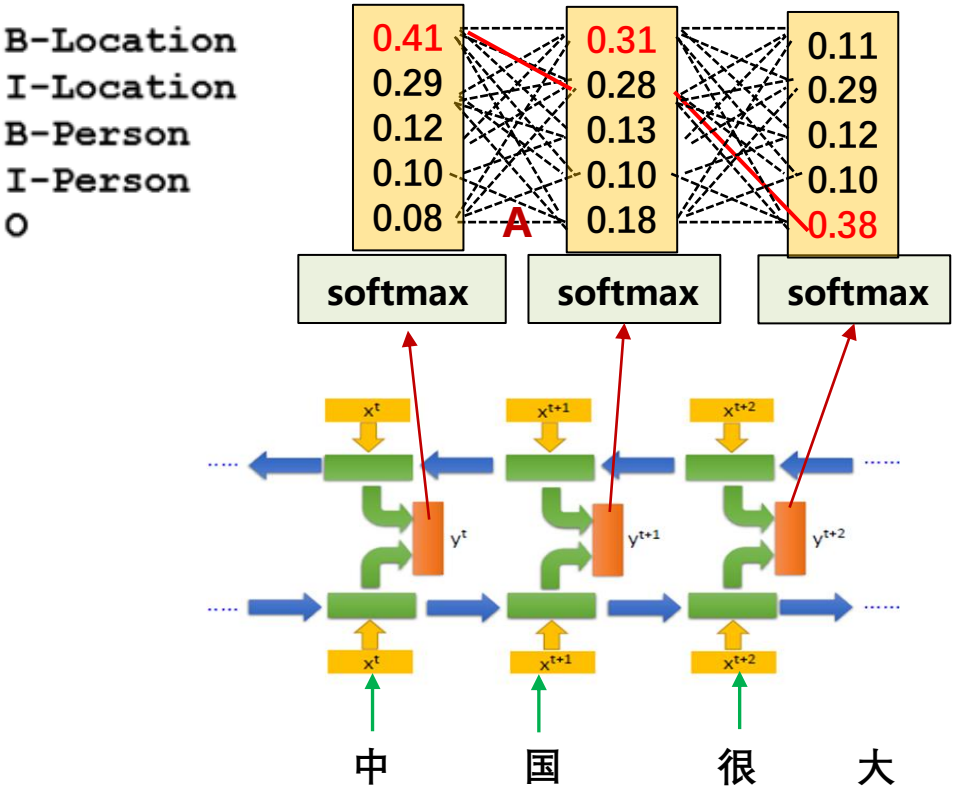
(2) 双向RNN+CRF 模型

对BiRNN+softmax 模型改进：

改进思路： 建立输出之间的关系

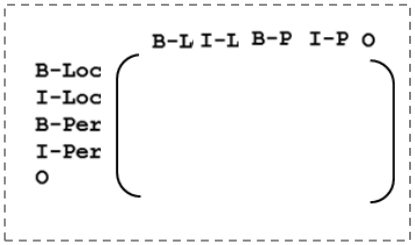


输出： 概率最大的序列　B-Loc　I-Loc　O



方法： 设一组参数A学习标签间的转移概率

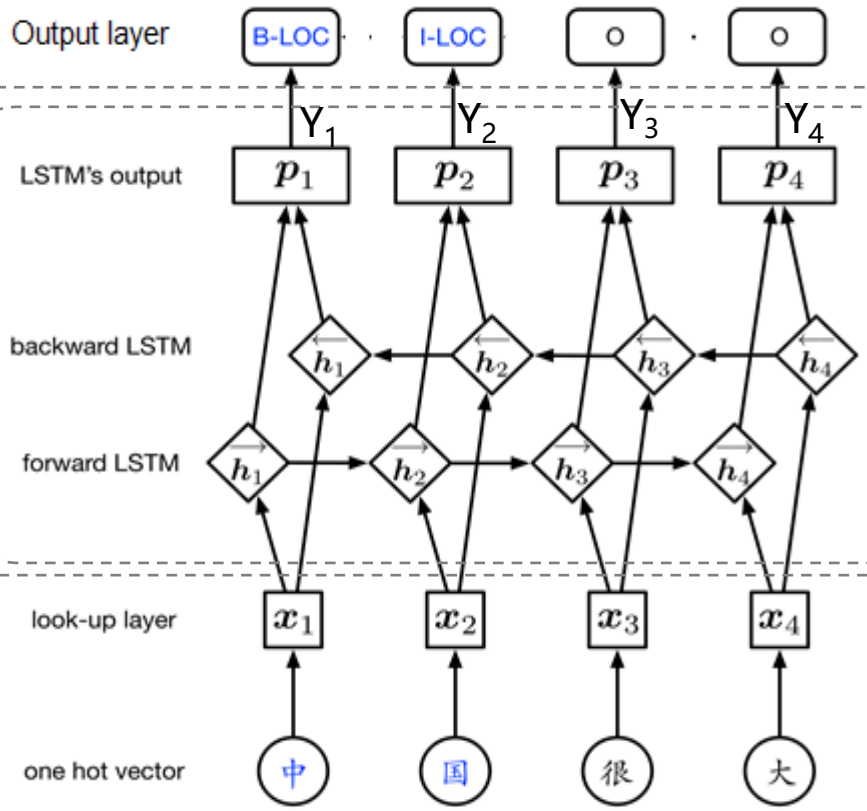
A： k× k方阵



K： 标签数

(2) 双向RNN+CRF 模型

如何改进BiRNN+softmax 模型（建立输出间联系）？



输出层： $Out_i = Max(y_i)$ 标签序列

$Y_i = softmax(p_i)$ (标签分布)

$p_i = \tanh(W_C[F(h_i), B(h_i)])$

反向隐藏层：

$$B(h_i) = \sigma((W2 \bullet X_i) + U2 \bullet B(h_{i+1}))$$

正向隐藏层：

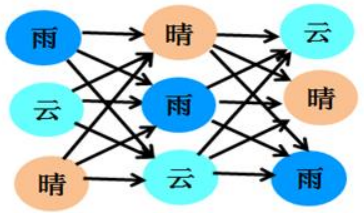
$$F(h_i) = \sigma((W1 \bullet X_i) + U1 \bullet F(h_{i-1}))$$

输入层： X_i ： 字 w_i 的字向量序列

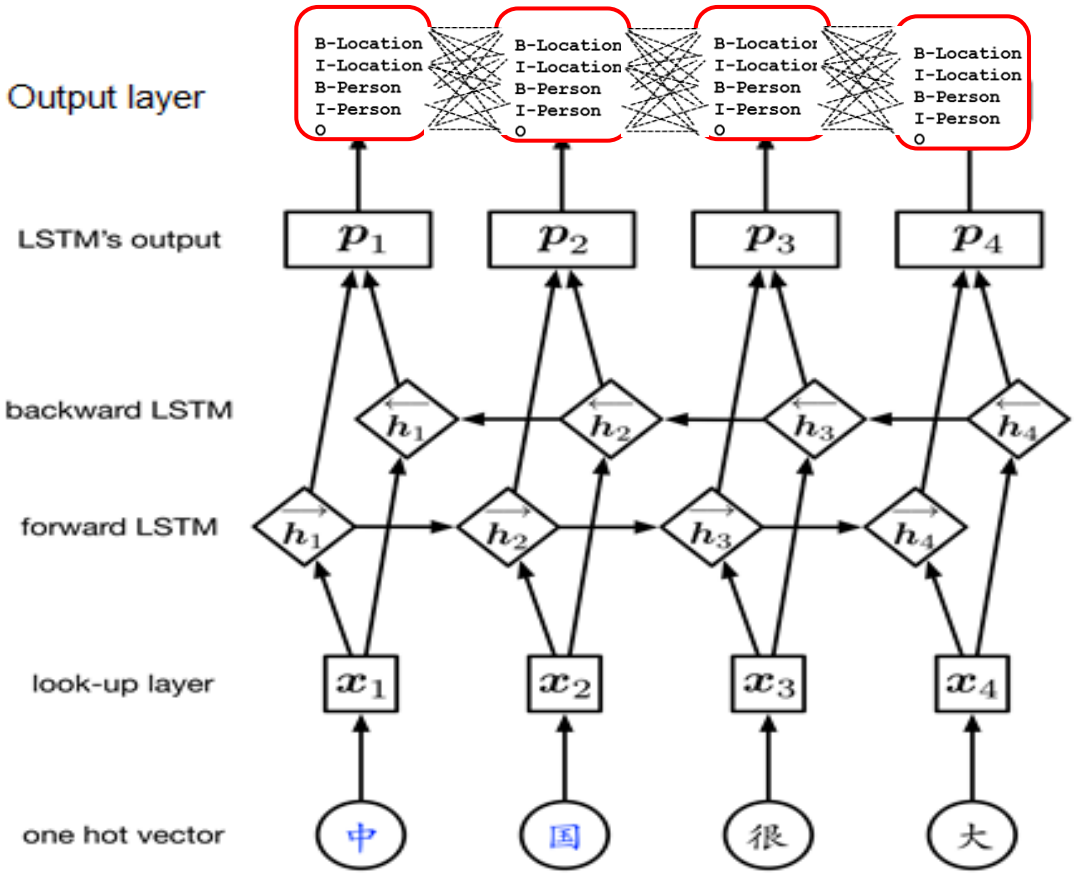
参数： W1,U1, W2, U2 ,Wc,

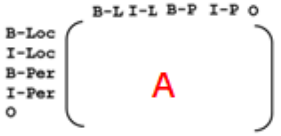
(2) 双向RNN+CRF 模型

如何改进BiRNN+softmax 模型（建立输出间联系）？



输出：概率最大的序列



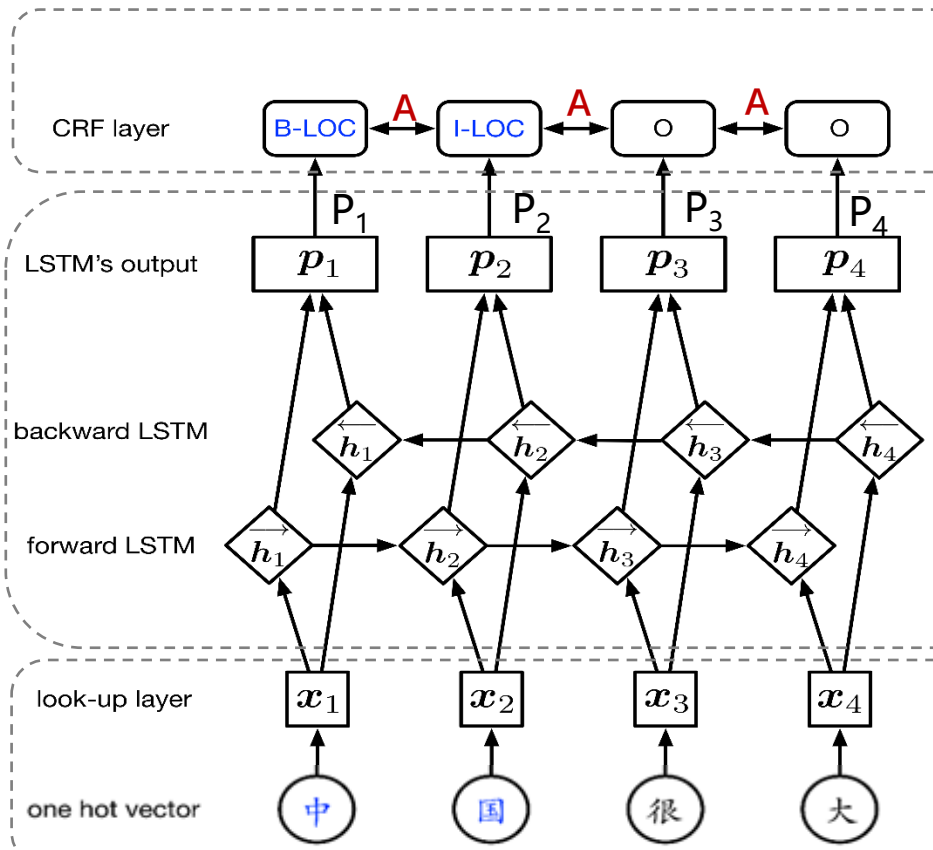


$$Y= \quad s(\mathrm{X},\mathrm{y}) = \sum_{i=0}^{\mathrm{L}} A_{y_i,y_{i+1}} + \sum_{i=1}^{\mathrm{L}} P_{i,y_i}$$

(2) 双向RNN+CRF 模型

(2) BiRNN+CRF 模型:

- **模型结构:**



输出层: $\mathbf{y}^* = \underset{\tilde{\mathbf{y}} \in \mathbf{Y_X}}{\operatorname{argmax}} s(\mathbf{X}, \tilde{\mathbf{y}})$.

Y= $s(\mathrm{X}, \mathrm{y}) = \sum_{i=0}^{\mathrm{L}} A_{y_i, y_{i+1}} + \sum_{i=1}^{\mathrm{L}} P_{i, y_i}$

P_i = softmax(p_i)

p_i =tanh($W_C[F(h_i), B(h_i)]$)

反向隐藏层:

$B(h_i) = \sigma((W2 \bullet \mathrm{X}_i) + U2 \bullet B(h_{i+1}))$

正向隐藏层:

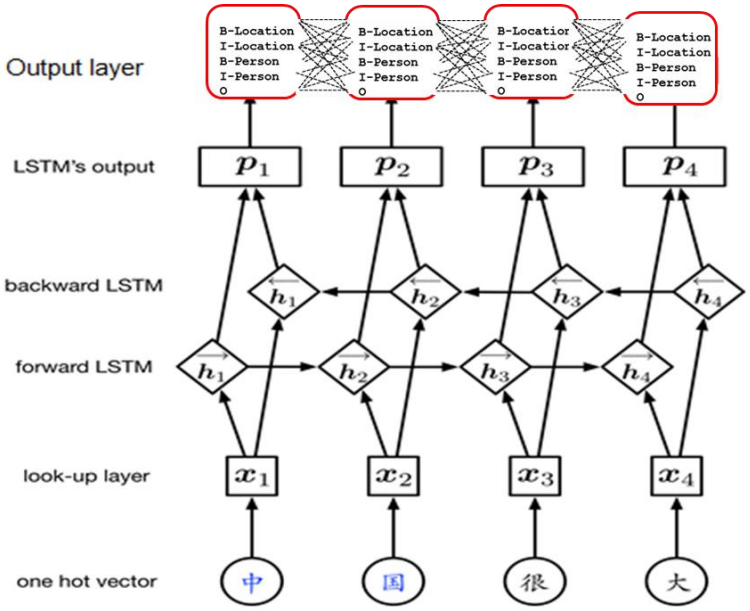
$F(h_i) = \sigma((W1 \bullet \mathrm{X}_i) + U1 \bullet F(h_{i-1}))$

输入层: X_i : 字w_i 的字向量

参数: W1,U1, W2, U2 ,Wc, **A**

(2) 双向RNN+CRF 模型

■ 模型学习 (有监督)



$\hat{Y}$ 格式： （ 0 0 0 … 1 0 … ）

$\hat{Y}$: B-PER I-PER O B-Loc

X: 张 三 在 北 京

如 标人名： 训练数据（有标注训练集）

张/B-PER 三/I-PER 在/O 北/B-Loc 京/I-Loc 旅/O 游/O …..

标签集：{B-PER， I-PER， B-Loc， I-Loc， O}

(2) 双向RNN+CRF 模型

■ 模型学习 (有监督)

- 损失函数：交叉熵损失

- 优化目标 $p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\tilde{\mathbf{y}}\in\mathbf{Y_X}} e^{s(\mathbf{X},\tilde{\mathbf{y}})}}.$ （ $s(\mathbf{X},\mathbf{y}) = \sum_{i=0}^{\mathrm{L}} A_{y_i,y_{i+1}} + \sum_{i=1}^{\mathrm{L}} P_{i,y_i}$ ）

  最大化 $\log(p(\mathbf{y}|\mathbf{X})) = s(\mathbf{X},\mathbf{y}) - \log\left(\sum_{\tilde{\mathbf{y}}\in\mathbf{Y_X}} e^{s(\mathbf{X},\tilde{\mathbf{y}})}\right) = s(\mathbf{X},\mathbf{y}) - \underset{\tilde{\mathbf{y}}\in\mathbf{Y_X}}{\text{logadd}}\, s(\mathbf{X},\tilde{\mathbf{y}})$

  其中，$\mathrm{Y_X}$是所有可能 的输出序列

- 用BPTT算法训练参数 θ=[A,W1,U1, V1, W2, U2, V2 ,Wc]

■ 模型预测： $\mathbf{y}^* = \underset{\tilde{\mathbf{y}}\in\mathbf{Y_X}}{\text{argmax}}\, s(\mathbf{X},\tilde{\mathbf{y}}).$

(2) 双向RNN+CRF 模型

■ **实验结果：**

Comparison of tagging performance on POS, chunking and NER tasks for various models.

		POS	CoNLL2000	CoNLL2003
Random	Conv-CRF (Collobert et al., 2011)	96.37	90.33	81.47
	LSTM	97.10	92.88	79.82
	BI-LSTM	97.30	93.64	81.11
	CRF	97.30	93.69	83.02
	LSTM-CRF	**97.45**	93.80	84.10
	BI-LSTM-CRF	97.43	**94.13**	**84.26**
Senna	Conv-CRF (Collobert et al., 2011)	97.29	94.32	88.67 (89.59)
	LSTM	97.29	92.99	83.74
	BI-LSTM	97.40	93.92	85.17
	CRF	97.45	93.83	86.13
	LSTM-CRF	97.54	94.27	88.36
	BI-LSTM-CRF	**97.55**	**94.46**	**88.83 (90.10)**

CRF效果好于只用LSTM或BI-LSTM

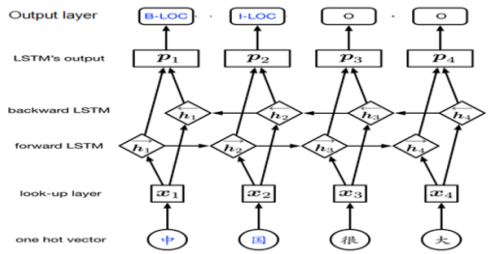
BI-LSTM与CRF结合的方法在多项任务上表现最好

双向RNN+softmax 与 双向RNN+CRF 模型对比：

如 标人名： 训练数据（有标注训练集） 标签集：{B-PER，I-PER，B-Loc，I-Loc， O}

张/B-PER 三/I-PER 在/O 北/B-Loc 京/I-Loc 旅/O 游/O …..

RNN+softmax：



$\hat{Y}$ 格式：(10000) (01000) (00000) (00100)

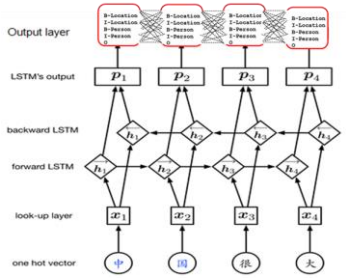
$\hat{Y}$：B-PER I-PER O B-Loc

X: 张 三 在 北 京

- 损失函数

交叉熵损失： $J(\theta;x,y) = -\sum_{j=1}^{k} y_j \log((y_{pred})_j)$ k 标签数

RNN+CRF



$\hat{Y}$ 格式： （ 0 0 0 … 1 0 … ）

$\hat{Y}$：B-PER I-PER O B-Loc

X: 张 三 在 北 京

- 损失函数：交叉熵损失
- 优化目标 $p(\mathrm{y}|\mathrm{X}) = \frac{e^{s(\mathrm{X},\mathrm{y})}}{\sum_{\tilde{\mathrm{y}}\in \mathrm{Y_X}} e^{s(\mathrm{X},\tilde{\mathrm{y}})}}$. （ $s(\mathrm{X},\mathrm{y}) = \sum_{i=0}^{\mathrm{L}} A_{y_i,y_{i+1}} + \sum_{i=1}^{\mathrm{L}} P_{i,y_i}$ ） $s(\mathrm{X},\mathrm{y}) = \sum_{i=0}^{\mathrm{L}} A_{y_i,y_{i+1}} + \sum_{i=1}^{\mathrm{L}} P_{i,y_i}$

参考文献：

http://wenku.baidu.com/view/3cf29130f111f18583d05a57.html

http://wenku.baidu.com/view/9121f528bd64783e09122b80.html

http://wenku.baidu.com/view/bbd57f82fc4ffe473268ab59.html?from=search

李航，统计学习方法　清华大学出版社

宗成庆，统计自然语言处理（第2版）

在 此 表 示 感 谢!

中国科学院大学
University of Chinese Academy of Sciences

谢 谢！

Thank you