



北京航空航天大學
BEIHANG UNIVERSITY

自然語言處理

人工智能研究院

主讲教师 沙磊

Contents

- 生成模型与判别模型
- 隐马尔科夫模型 HMM
- 条件随机场模型 CRF

生成模型与判别模型

- Generative Models (Two-step)
 - Infer class-conditional densities $p(x|C_k)$ and priors $p(C_k)$
 - then use Bayes theorem to determine posterior probabilities.

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

- Discriminative Models (One-step)
 - Directly infer posterior probabilities $p(C_k|x)$

生成模型

- Given M variables, $x = (x_1, \dots, x_M)$, class variable y and joint distribution $p(x,y)$ we can

- Marginalize

$$p(y) = \sum_x p(x, y)$$

- Condition

$$p(y|x) = \frac{p(x, y)}{p(x)}$$

- By conditioning the joint pdf we form a classifier

- Huge need for samples

- If x_i are binary, need 2^M values to specify $p(x,y)$

ML方法分类

- Generative Methods
 - “Generative” since sampling can generate synthetic data points
 - Popular models
 - Naïve Bayes, Mixtures of multinomials
 - Mixtures of Gaussians, Hidden Markov Models
 - Bayesian networks, Markov random fields
- Discriminative Methods
 - Focus on given task – better performance
 - Popular models
 - Logistic regression, SVMs
 - Traditional neural networks, Nearest neighbor
 - Conditional Random Fields (CRF)



隐马尔科夫 模型

隐马尔科夫模型

- 隐马尔科夫模型(Hidden Markov Model, HMM)是对马尔科夫模型的一种扩充。
- 隐马尔科夫模型的基本理论形成于上世纪60年代末期和70年代初期。(L.E.Baum)
- 70年代, CMU的J.K.Baker以及IBM 的F.Jelinek 等把隐马尔科夫模型应用于语音识别。
- 隐马尔科夫模型在计算语言学中有着广泛的应用。例如隐马尔科夫模型在词类自动标注中的应用。

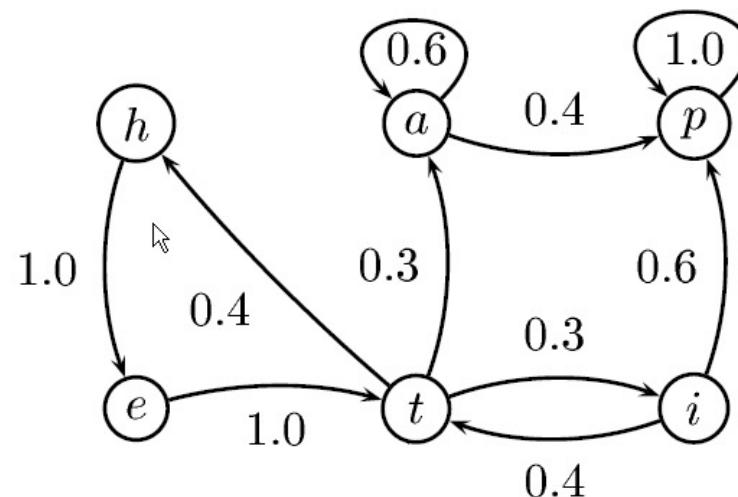
马尔科夫模型

- 马尔科夫模型是由Andrei A. Markov于1913年提出的。
- 设 S 是一个由有限个状态组成的集合。
 - $S=\{1, 2, 3, \dots, n-1, n\}$
- 随机序列 X 在 t 时刻所处的状态为 q_t , 其中 $q_t \in S$, 若有:
 - $P(q_t = j | q_{t-1} = i, q_{t-2} = k, \dots) = P(q_t = j | q_{t-1} = i)$
 - 则随机序列 X 构成一个一阶马尔科夫链。 (Markov Chain)
- 令 $P(q_t = j | q_{t-1} = i) = P(q_s = j | q_{s-1} = i)$, 则对于所有的 i, j 有下面的关系成立:

$$a_{ij} = P(q_t = j | q_{t-1} = i) \quad 1 \leq i, j \leq n \quad \sum_{j=1}^n a_{ij} = 1 \quad a_{ij} \geq 0$$

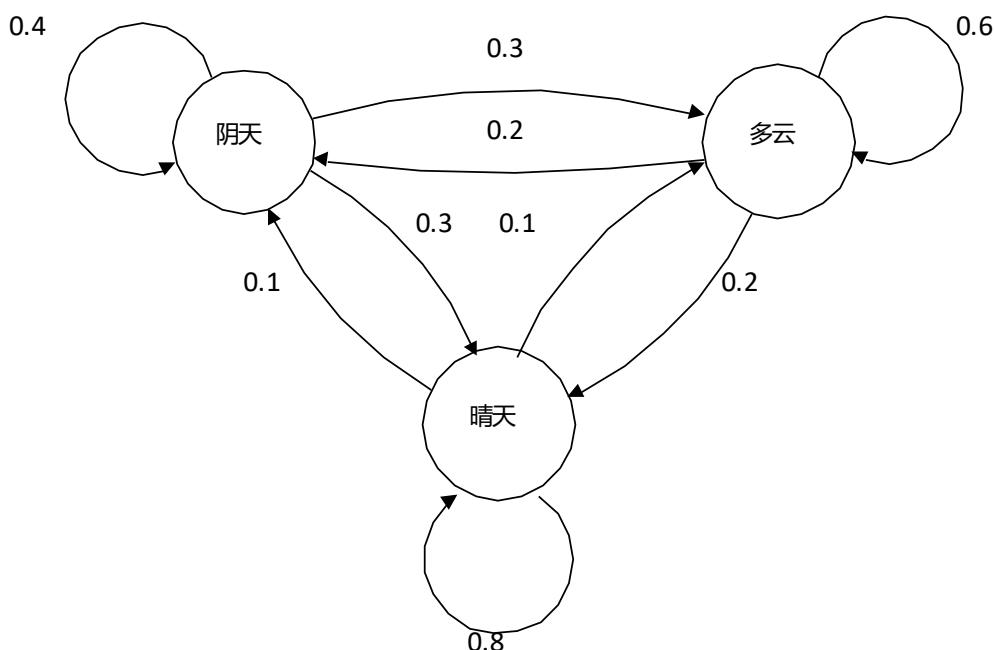
马尔科夫模型

- 一阶马尔科夫模型可以描述为一个二元组(S, A)， S 是状态的集合，而 A 是所有状态转移概率组成的一个 n 行 n 列的矩阵，其中每一个元素 a_{ij} 为从状态 i 转移到状态 j 的 概率。
- 同有限状态自动机类似，状态转移关系也可以用状态转换图来表示。



马尔科夫模型举例

- 天气的变化，三种状态{1(阴天), 2(多云), 3(晴天)}。
- 今天的天气情况仅和昨天的天气状况有关。
- 根据对历史数据的观察得到下列状态转移关系。



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

马尔科夫模型

- 如果把晴天称为状态3的输出，阴天称为状态1 的输出，多云称为状态2的输出。 因为状态和输出是一对一的关系，所以根据观察到的输出序列就可以决定模型中的状态转换序列。
- 对于马尔科夫模型，给定了观察序列，同时也确定了状态转换序列。

例如有关天气状况的观察序列。

(晴 晴 晴 阴 阴 晴 云 晴)

则状态转换序列为

(3, 3, 3, 1, 1, 3, 2, 3)

坛子与小球

- 在一个房间中，假定有 N 个坛子，每个坛子中都装有不同颜色的小球，并且假定总共有 M 种不同颜色的小球。
- 一个精灵在房间中首先随机地选择一个坛子，再从这个坛子中随机选择一个小球，并把小球的颜色报告给房间外面的人员记录下来作为观察值。
- 精灵然后把球放回到坛子中，以当前的坛子为条件再随机选择一个坛子，从中随机选择一个小球，并报告小球的颜色，然后放回小球，如此继续...，随着时间的推移，房间外的人会得到由这个过程产生的一个小球颜色的序列。

坛子与小球

- 如果令每一个坛子对应与一个状态，令小球颜色对应状态的输出。
- 可以用一个一阶马尔科夫过程来描述坛子的选择过程。
- 在马尔科夫过程中，每个状态只有一个输出，但在坛子和小球的问题中。可以从每个坛子中拿出不同颜色的小球。也就是每个状态能按照特定的概率分布产生多个输出，状态和输出之间不再是一一对应关系。
- 在坛子与小球问题中，如果给定一个观察序列(不同颜色的小球序列)，不能直接确定状态转换序列(坛子的序列)，因为状态转移过程被隐藏起来了。所以这类随机过程被称为隐马尔科夫过程。

隐马尔科夫模型

- 隐马尔可夫模型 λ 可以表示为一个五元组(S, V, A, B, π)
 - ❖ S 是一组状态的集合。
 - ❖ $S = \{1, 2, 3, \dots, N\}$ (状态 n 对应坛子 n)
 - ❖ V 是一组输出符号组成的集合。
 - ❖ $V = \{v_1, v_2, v_3, \dots, v_M\}$ (v_1 对应红色小球)
 - ❖ A 是状态转移矩阵， N 行 N 列。
 - ❖ $A = [a_{ij}]$
 - ❖ $a_{ij} = P(q_{t+1}=j \mid q_t=i), 1 \leq i, j \leq N$

隐马尔科夫模型

- ❖ B 是输出符号的概率分布。
- ❖ $B = \{ b_j(k) \}$ $b_j(k)$ 表示在状态 j 时输出符号 v_k 的概率
- ❖ $b_j(k) = P(v_k | j), 1 \leq k \leq M, 1 \leq j \leq N$

- ❖ π 是初始状态概率分布 $\pi = \{ \pi_i \}$
- ❖ $\pi_i = P(q_1 = i)$ 表示时刻 1 选择某个状态的概率。

- 隐马尔可夫过程是一个双重随机过程，其中一重随机过程不能直接观察到，通过状态转移概率矩阵描述。另一重随机过程输出可以观察到的观察符号，这由输出概率来定义。

利用隐马尔科夫模型生成观察序列

- 可以把隐马尔可夫模型看做一个符号序列的生成装置，按照一定的步骤，隐马尔可夫模型可以生成下面的符号序列：
- $O = (o_1 o_2 o_3 \dots o_T)$

1. 令 $t = 1$ ，按照初始状态概率分布 π 选择一个初始状态 $q_1 = i$ 。
2. 按照状态 i 输出符号的概率分布 $b_i(k)$ 选择一个输出值 $o_t = v_k$ 。
3. 按照状态转移概率分布 a_{ij} 选择一个后继状态 $q_{t+1} = j$ 。
4. 若 $t < T$ ，令 $t = t + 1$ ，并且转移到算法第2步继续执行，否则结束。

抛掷硬币

- 三枚硬币，随机选择一枚，进行抛掷，记录抛掷结果。可以描述为一个三个状态的隐马尔科夫模型 λ 。

- $\lambda = (S, V, A, B, \pi)$, 其中

- $S = \{1, 2, 3\}$

- $V = \{H, T\}$

A 如下表所示

	1	2	3
1	0.9	0.05	0.05
2	0.45	0.1	0.45
3	0.45	0.45	0.1

B 如下表所示

	1	2	3
H	0.5	0.75	0.25
T	0.5	0.25	0.75

$$\pi = \{1/3, 1/3, 1/3\}$$

抛掷硬币

- 问题一：

给定上述模型，观察到下列抛掷结果的概率是多少？

$$O = (H H H H T H T T T)$$

- 问题二：

给定上述模型，若观察到上述抛掷结果，最可能的硬币选择序列
(状态转换序列)是什么？

- 问题三：

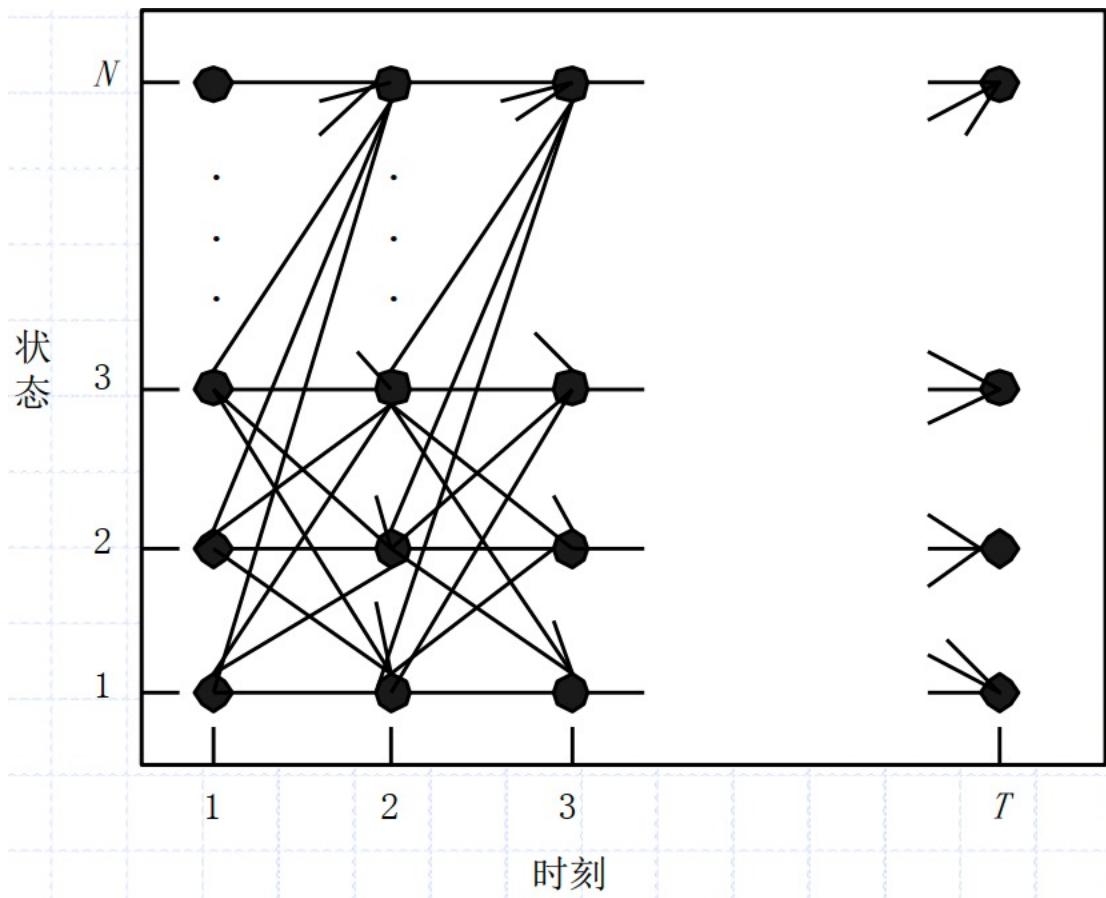
若上述模型中的状态转移矩阵 A 、状态输出概率 B 和初始状态分布 π 均未知，如何根据观察序列得到它们？

隐马尔科夫模型的三个问题

- 给定HMM $\lambda = (A, B, \pi)$, 给定观察序列 $O = (o_1 o_2 o_3 \dots o_T)$, 如何有效地计算出观察序列的概率, 即 $P(O|\lambda)$? (估算问题) (另一种语言模型)
- 给定HMM $\lambda = (A, B, \pi)$, 给定观察序列 $O = (o_1 o_2 o_3 \dots o_T)$, 如何寻找一个状态转换序列 $q = (q_1 q_2 q_3 \dots q_T)$, 使得该状态转换序列最有可能产生上述观察序列? (解码问题)
- 在模型参数未知或不准确的情况下, 如何根据观察序列 $O = (o_1 o_2 o_3 \dots o_T)$ 求得模型参数或调整模型参数。按照 MLE 的原则, 即如何确定一组模型参数, 使得 $P(O|\lambda)$ 最大? (学习问题或训练问题)

问题1：估算观察序列概率

- 对隐马尔可夫模型而言，状态转换序列是隐藏的，一个观察序列可能由任何一种状态转换序列产生。因此要计算一个观察序列的概率值，就必须考虑所有可能的状态转换序列。
- 右图表示了产生观察序列 $O = (o_1 o_2 o_3 \dots o_T)$ 的所有可能的状态转换序列。



估算观察序列概率

- 给定 λ , 以及状态转换序列 $q = (q_1 q_2 q_3 \dots q_T)$ 产生观察序列 $O = (o_1 o_2 o_3 \dots o_T)$ 的概率可以通过下面的公式计算:

$$P(O|q, \lambda) = b_{q_1}(o_1)b_{q_2}(o_2)\dots b_{q_T}(o_T)$$

- 给定 λ , 状态转换序列 $q = (q_1 q_2 q_3 \dots q_T)$ 的概率可以通过下面的公式计算:

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

- 则 O 和 q 的联合概率为: $P(O, q|\lambda) = P(O|q, \lambda)P(q|\lambda)$

- 考虑所有的状态转换序列, 则

$$P(O|\lambda) = \sum_q P(O, q|\lambda) = \sum_{q_1 \dots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T)$$

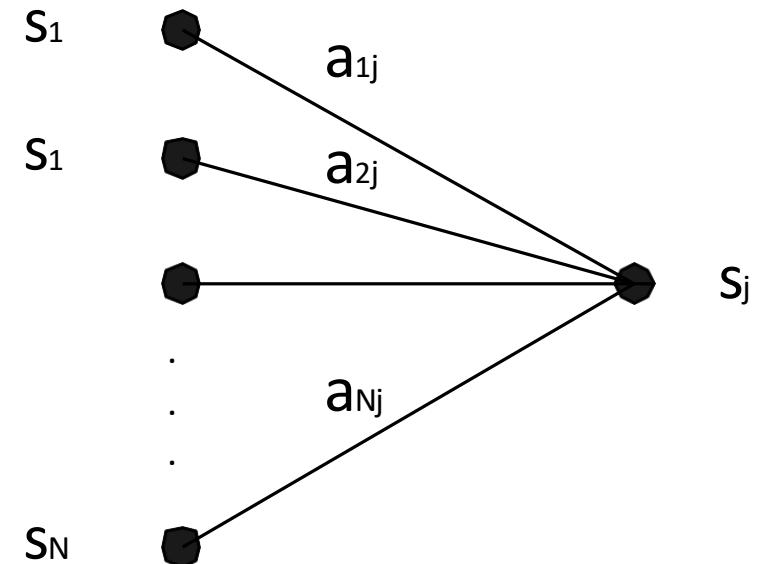
估算观察序列概率

- 理论上，可以通过穷举所有状态转换序列的办法计算观察序列 O 的概率。
- 实际上，这样做并不现实。
 - 可能的状态转换序列共有 N^T 个。
 - 需要做 $(2T-1)N^T$ 次乘法运算， N^T-1 次加法运算。
 - 若 $N=5$, $T=100$, 则 $(2 \times 100-1) \times 5^{100} \approx 10^{72}$
- 需要寻找更为有效的计算方法。

向前算法(Forward Algorithm)

- 向前变量 $\alpha_t(i)$ $\alpha_t(i) = P(o_1 o_2 o_3 \dots o_t, q_t = i | \lambda)$
- $\alpha_t(i)$ 的含义是，给定模型 λ ，时刻 t ，处在状态 i ，并且部分观察序列 $o_1 o_2 o_3 \dots o_t$ 的概率。
- 显然有 $\alpha_1(i) = \pi_i b_i(o_1) (1 \leq i \leq N)$
- 若 $\alpha_t(i) (1 \leq i \leq N)$ 已知，如何计算 $\alpha_{t+1}(i)$ ？

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), 1 \leq t \leq T-1, 1 \leq j \leq N$$



向前算法

1. 初始化 $\alpha_1(i) = \pi_i b_i(o_1) (1 \leq i \leq N)$

2. 迭代计算

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), 1 \leq t \leq T-1, 1 \leq j \leq N$$

3. 终止

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

◆ 计算量

- $N(N+1)(T-1)+N$ 次乘法
- $N(N-1)(T-1)$ 次加法
- 若 $N=5, T=100$, 则
大约需要 5000 次运算

计算实例

- 抛掷硬币问题，计算观察到(HHT)的概率。

$\alpha_t(i)$	H	H	T	$P(HHT \lambda)$
1	0.16667	0.15000	0.08672	
2	0.25000	0.05312	0.00684	0.11953
3	0.08333	0.03229	0.02597	

向后算法 (Backward Algorithm)

- 向后变量 $\beta_t(i) \quad \beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = i, \lambda)$
- $\beta_t(i)$ 的含义是，在给定模型 λ ，时刻 t ，处在状态 i ，并且部分观察序列为 $o_{t+1} o_{t+2} \dots o_T$ 的概率。

$$\beta_T(i) = 1$$

- 若 $\beta_{t+1}(j) (1 \leq j \leq N)$ 已知，如何计算 $\beta_t(i)$

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

向后算法

1. 初始化 $\beta_T(i) = 1 (1 \leq i \leq N)$

2. 迭代计算

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

3. 终止

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(o_1) \beta_1(i)$$

计算实例

- 抛掷硬币问题，计算观察到 $(H\ H\ T)$ 的概率。

	H	H	T		$P(H\ H\ T \lambda)$
$\beta_t(i)$	$\pi_i b_i(H) \beta_1(i)$	$\beta_1(i)$	$\beta_2(i)$	$\beta_3(i)$	
1	0.04203	0.25219	0.50000	1.00000	
2	0.05074	0.20297	0.58750	1.00000	0.11953
3	0.02676	0.32109	0.41250	1.00000	

求解最佳状态转换序列

- 隐马尔可夫模型的第二个问题是计算出一个能最好解释观察序列的状态转换序列。
- 理论上，可以通过枚举所有的状态转换序列，并对每一个状态转换序列 q 计算 $P(O, q | \lambda)$ ，能使 $P(O, q | \lambda)$ 取最大值的状态转换序列 q^* 就是能最好解释观察序列的状态转换序列，即：

$$q^* = \arg \max_q P(O, q | \lambda)$$

- 同样，这不是一个有效的计算方法，需要寻找更好的计算方法。

韦特比算法(Viterbi Algorithm)

- 韦特比变量 $\delta_t(i)$

$$\delta_t(i) = \max_{q_1 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1}, q_t = i, o_1 o_2 \dots o_t | \lambda)$$

- $\delta_t(i)$ 的含义是，给定模型 λ ，在时刻 t 处于状态 i ，观察到 $o_1 o_2 o_3 \dots o_t$ 的最佳状态转换序列为 $q_1 q_2 \dots q_t$ 的概率。

$$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$$

- 若 $\delta_t(i) (1 \leq i \leq N)$ 已知，如何计算 $\delta_{t+1}(i)$ ？

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(o_{t+1})$$

- 如何记录路径？设定 T 个数组 $\psi_1(N), \psi_2(N), \dots, \psi_T(N)$

$\psi_t(i)$ 记录在时刻 t 到达状态 i 的最佳状态转换序列 $t-1$ 时刻的最佳状态。

韦特比算法

1. 初始化 $\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$
 $\psi_1(i) = 0$

2. 迭代计算

$$\delta_t(j) = [\max_{1 \leq i \leq N} \delta_{t-1}(i)a_{ij}]b_j(o_t) \quad \psi_t(j) = \arg \max_{1 \leq i \leq N} \delta_{t-1}(i)a_{ij}]$$

3. 终止

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

4. 求解最佳路径

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, \dots, 1$$

计算实例

- 抛掷硬币问题，观察到 (HHT) ，寻找产生该观察序列的最佳路径以及最佳路径的概率。

$\delta_t(i)$	H	H	T	P^*
1	0.16667	0.07500	0.03375	
2	0.25000	0.02812	0.00316	0.03375
3	0.08333	0.02812	0.00949	

$\psi_t(i)$	$\psi_1(i)$	$\psi_2(i)$	$\psi_3(i)$	q^*
1	0	1	1	
2	0	3	3	1
3	0	2	2	

- 最佳状态转换序列为1 1 1

参数学习

- 隐马尔科夫模型的第三个问题是根据观察序列 $O = (o_1 o_2 o_3 \dots o_T)$ 求得模型参数或调整模型参数，即如何确定一组模型参数使得 $P(O|\lambda)$ 最大？
- 隐马尔科夫模型的前两个问题均假设模型参数已知，第三个问题是模型参数未知，求最佳模型的问题，是三个问题中最为困难的问题。

有指导的参数学习 (supervised learning)

- 在模型(λ)未知的情况下，如果给定观察序列的同时，也给定了状态转换序列，此时可以通过有指导的学习方法学习模型参数。例如给定下面的训练数据，可以通过最大似然估计法估计模型参数：

$H/1 H/1 T/1 T/2 H/3 T/5 \dots$

$T/2 H/1 T/2 H/3 H/3 H/1 \dots$

- 参数学习非常简单，在训练数据足够大的前提下，效果不错。
- 缺点，状态信息未知时无法使用。或者要由人工标注状态信息，代价高。
- 在NLP中，在无指导学习效果不佳时，需要采用有指导学习。

无指导的参数学习 (unsupervised learning)

- 在模型(λ)未知的情况下，如果仅仅给定了观察序列，此时学习模型的方法被称做无指导的学习方法。
- 对于隐马尔科夫模型而言，采用无指导学习方法，没有解析方法。通常要首先给定一组不准确的参数，再通过反复迭代逐步求精的方式调整模型参数，最终使参数稳定在一个可以接受的精度。
- 利用无指导的学习方法估计隐马尔科夫模型参数时，并不能一定保证求得最优模型，一般能得到一个局部最优模型。

直观的想法

- 给定一组初始参数(A B π)
- 由于没有给定状态转换序列，无法计算状态转移的频率、状态输出的频率以及初始状态频率。
- 假定任何一种状态转换序列都可能。
- 对每种状态转换序列中的频次加权处理，计算状态转移、状态输出、以及初始状态的期望频次
- 利用计算出的期望频次更新 A 、 B 和 π

直观的想法

- 权值如何选择

对状态转换序列 q 而言，选择 $P(q|O, \lambda)$

- 理论上可行，现实不可行

要考虑所有的状态转移路径

需要多次迭代，问题更为严重

- 需要更为有效的算法，即Baum-Welch算法

Baum-Welch Algorithm

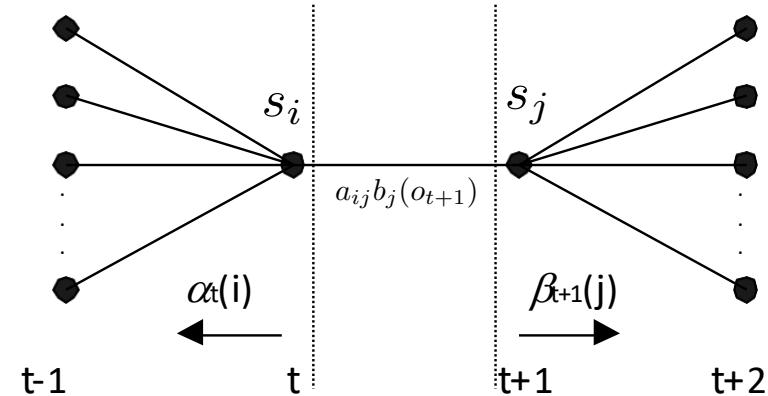
- 定义变量 $\xi_t(i, j)$

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

- $\xi_t(i, j)$ 含义是，给定模型 λ 和观察序列 O ，在时刻 t 处在状态 i ，时刻 $t+1$ 处在状态 j 的期望概率。

- $\xi_t(i, j)$ 可以进一步写成：

$$\begin{aligned}
 \xi_t(i, j) &= \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)} \\
 &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \\
 &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}
 \end{aligned}$$



Baum-Welch Algorithm

- 定义变量 $\gamma_t(i)$, 令其表示在给定模型以及观察序列的情况下, t 时刻处在状态 i 的概率, 则有:

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

- 观察序列 O 中从状态 i 出发的转换的期望概率

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

- 观察序列 O 中从状态 i 到状态 j 的转换的期望概率

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

Baum-Welch Algorithm

- 关于 π, A, B , 一种合理的估计方法如下

$$\bar{\pi}_i = \gamma_1(i)$$

在 $t = 1$ 时处在状态 i 的期望概率

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

从状态 i 到状态 j 的转换的期望概率除以从状态 i 出发的转换的期望概率

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j) \delta(x_t, v_k)}{\sum_{t=1}^T \gamma_t(j)}$$

当 $o_t = v_k$ 时 , $\delta(o_t, v_k) = 1$
当 $o_t \neq v_k$ 时 , $\delta(o_t, v_k) = 0$

在状态 j 观察到 v_k 的期望概率

处在状态 j 的期望概率

Baum-Welch Algorithm

- 利用上述结论，即可进行模型估算
- 选择模型参数初始值，初始值应满足隐马尔科夫模型的要求，即：

$$\sum_{i=1}^N \pi_i = 1 \quad \sum_{j=1}^N a_{ij} = 1, 1 \leq i \leq N \quad \sum_{k=1}^M b_j(k) = 1, 1 \leq j \leq N$$

- 将初始值代入前面的公式中，计算一组新的参数 $\bar{\pi}, \bar{A}, \bar{B}$
- 再将新的参数代入，再次计算更新的参数。
- 如此反复，直到参数收敛。

Baum-Welch Algorithm

- Baum-Welch算法是一种EM算法。
- E-step:
 - 计算 $\xi_t(i, j)$ 和 $\gamma_t(i)$
- M-step:
 - 估计模型 λ
- 终止条件

$$\left| \log(P(O|\lambda_{i+1})) - \log(P(O|\lambda_i)) \right| < \epsilon$$

Baum-Welch Algorithm

- Baum等人证明要么估算值 $\bar{\lambda}$ 和估算前的参数值 λ 相等，要么估算值 $\bar{\lambda}$ 比估算前的参数值 λ 更好的解释了观察序列 O 。
- 参数最终的收敛点并不一定是一个全局最优值，但一定是一个局部最优值。

L.R.Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech recognition, Proc. IEEE, 77(2): 257-286, 1989

隐马尔科夫模型的实现

- 浮点溢出问题
 - 对于韦特比算法，采用取对数的方式
 - 对于Baum-Welch算法，采用放大因子
 - 对于向前算法采用放大因子以及取对数的方式。

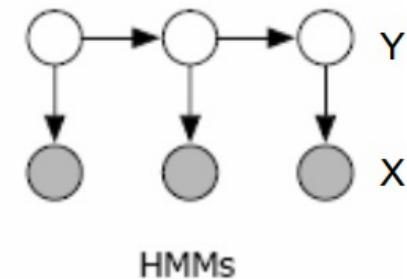


CRF

序列模型

- Hidden Markov Model (HMM)

$$p(y, x) = \prod_{t=1}^T p(y_t | y_{t-1}) p(x_t | y_t)$$

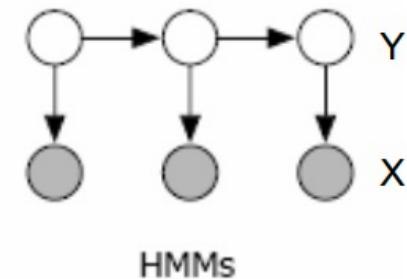


- 独立性假设：
 - 每个状态只直接依赖于其前一个
 - 每个观察变量只依赖于当前状态
- 局限性：
 - 观察变量X之间存在强独立性假设。
 - 建模联合概率 $p(y, x)$ 引入大量参数，这需要建模分布 $p(x)$

序列模型

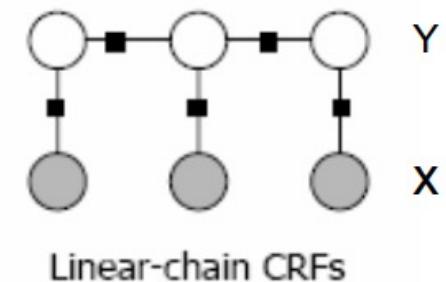
- Hidden Markov Model (HMM)

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^T p(y_t | y_{t-1}) p(x_t | y_t)$$



- Conditional Random Fields (CRF)

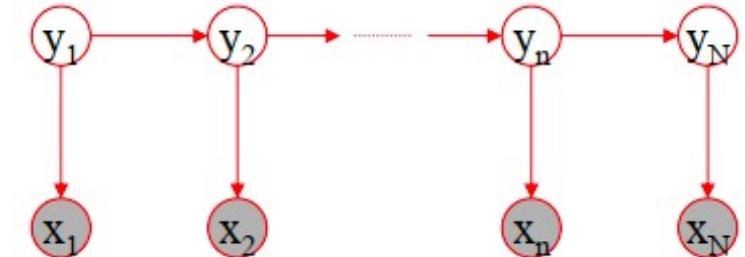
$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$$



- 条件随机场 (CRF) 的一个重要优势是它们具有很大的灵活性，可以包含各种任意的、非独立的观测特征。

Generative Model: HMM

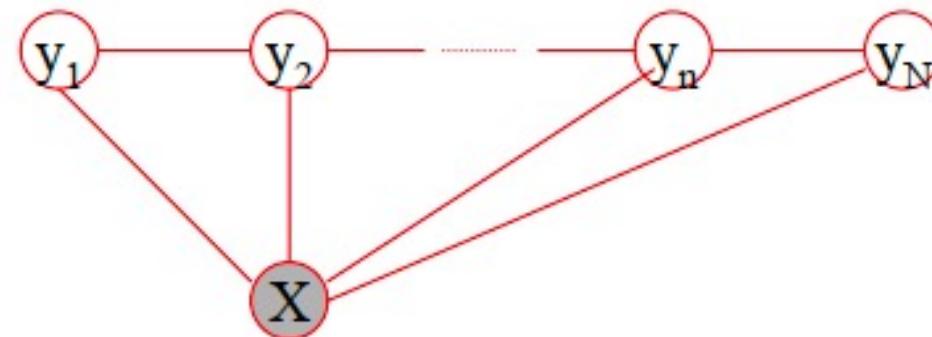
- X is observed data sequence to be labeled,
- Y is the random variable over the label sequences
- HMM is a distribution that models $p(Y, X)$
- Joint distribution is
- Highly structured network indicates conditional independences,
 - Past states independent of future states
 - Conditional independence of observed given its state.



$$p(\mathbf{Y}, \mathbf{X}) = \prod_{n=1}^N p(y_n | y_{n-1}) p(x_n | y_n)$$

针对序列模型的判别模型

- CRF模型建模了给定观测值 X 条件下的条件分布 $p(Y|X)$
- CRF是一个随机场，全局地以观测值 X 为条件
- 从联合分布 $p(Y,X)$ 中得出的条件分布 $p(Y|X)$ 可以被重写为一个**马尔可夫随机场**。



Markov Random Field (MRF)

- 也称为无向图模型
- 变量集 x 的联合分布由一个无向图定义
- 其中 C 是最大团 (clique) (每个节点与每个其他节点相连)
 - x_C 是该团中的变量集, ψ_C 是潜在函数potential function (或局部函数(local function)或兼容函数(compatibility function)) , 满足 $\psi_C(x_C) > 0$, 通常 $\psi_C(x_C) = \exp\{-E(x_C)\}$, 而 Z 是用于归一化的配分函数。
 - 模型是指一组分布, 而场则指一个具体的分布。

$$Z = \sum_x \prod_C \psi_C(x_C)$$

$$p(x) = \frac{1}{Z} \prod_C \psi_C(x_C)$$

MRF with Input-Output Variables

- X 是一组被观测到的输入变量
 - X 的元素用 x 表示
- Y 是一组我们要预测的输出变量
 - Y 的元素用 y 表示
- A 是 $X \cup Y$ 的子集
 - A 中属于 $A \cap X$ 的元素用 x_A 表示
 - A 中属于 $A \cap Y$ 的元素用 y_A 表示
- 那么无向图模型的形式为

$$p(x,y) = \frac{1}{Z} \prod_A \Psi_A(x_A, y_A)$$

where

$$Z = \sum_{x,y} \prod_A \Psi_A(x_A, y_A)$$

MRF Local Function

- 假设每个局部函数的形式为

$$\Psi_A(x_A, y_A) = \exp \left\{ \sum_m \theta_{Am} f_{Am}(x_A, y_A) \right\}$$

- 其中 θ_A 是一个参数向量， f_A 是特征函数， $m=1..M$ 是特征下标。

From HMM to CRF

- HMM 中

$$p(Y, X) = \prod_{n=1}^N p(y_n | y_{n-1}) p(x_n | y_n)$$

- 可以被写作：

分布参数： $\theta = \{\lambda_{ij}, \mu_{oi}\}$

$$p(Y, X) = \frac{1}{Z} \exp \left\{ \sum_n \sum_{i,j \in S} \lambda_{ij} \mathbb{I}_{y_n=i} \mathbb{I}_{y_{n-1}=j} + \sum_n \sum_{i \in S} \sum_{o \in O} \mu_{oi} \mathbb{I}_{y_n=i} \mathbb{I}_{x_n=o} \right\}$$

- 进一步写作：

$$p(Y, X) = \frac{1}{Z} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}$$

特征函数有如下形式： $f_m(y_n, y_{n-1}, x_n)$
对每个状态转换 $i \rightarrow j$ 需要一个特征

$$f_{ij}(y, y', x) = \mathbb{I}(y = i) \mathbb{I}(y' = j)$$

对每个状态-观察对也需要一个特征

$$f_{io}(y, y', x) = \mathbb{I}(y = i) \mathbb{I}(x = o)$$

From HMM to CRF

$$p(Y, X) = \frac{1}{Z} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}$$

- 进一步

$$p(Y|X) = \frac{p(y, x)}{\sum_{y'} p(y', x)} = \frac{\exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}}{\sum_{y'} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y'_n, y'_{n-1}, x_n) \right\}}$$

CRF definition

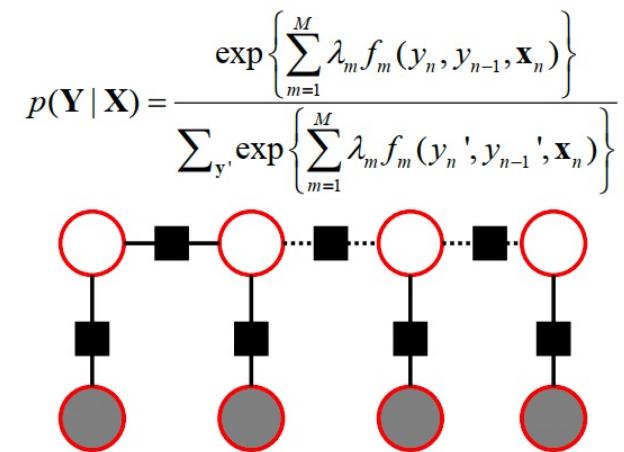
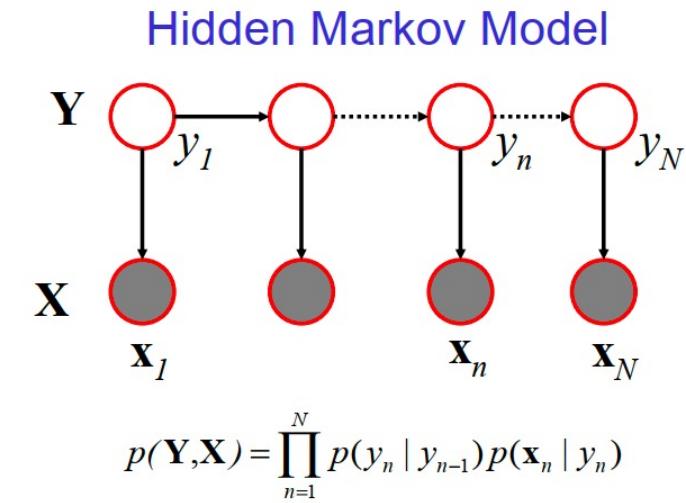
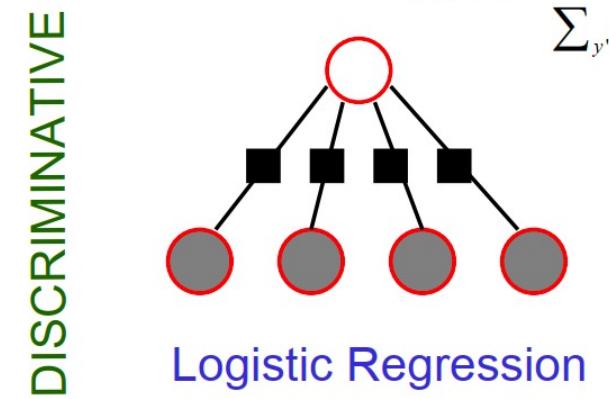
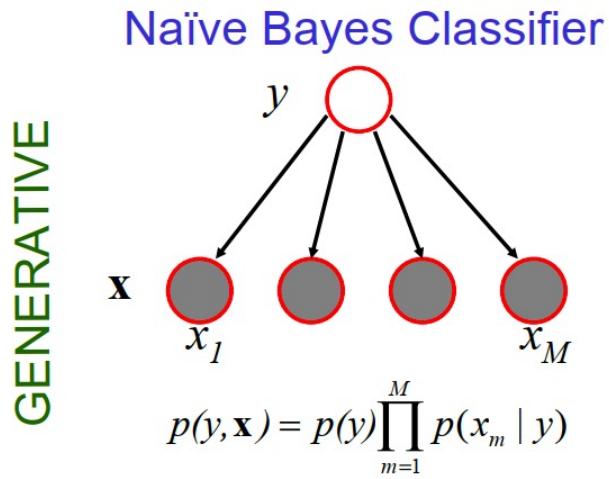
- 线性链CRF定义为分布 $p(Y|X)$, 其形式如下:

$$p(Y|X) = \frac{1}{Z(X)} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}$$

- 其中 $Z(X)$ 是一个实例特定的归一化函数。

$$Z(X) = \sum_{y'} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y'_n, y'_{n-1}, x_n) \right\}$$

功能模型



CRF的优势

- CRF放松了对于给定标签的观测数据的条件独立性的假设
- CRF可以包含任意的特征函数
 - 每个特征函数可以使用整个输入数据序列。 观测数据片段的标签概率可能取决于任何过去或未来的数据片段。
- CRF可以避免其他具有偏向后继状态较少的状态的判别性
马尔可夫模型的限制

CRF 优化

$$p(Y|X) = \frac{1}{Z(X)} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}$$

- 目标函数（极大似然法）

$$\max_{\lambda_m \in \mathbb{R}^+} \prod_{x,y} p(y|x; \lambda)^{\tilde{P}(x,y)}$$

- Why not set the loss as this?

$$\tilde{P}(x, y) = \tilde{P}(y|x)\tilde{P}(x)$$

$$\max_{\lambda_m \in \mathbb{R}^+} \prod_{x,y} p(y|x; \lambda)^{\tilde{P}(y|x)}$$

CRF 优化

- 实际依然采用对数值进行优化

$$\min_{\lambda \in \mathbb{R}^+} f(\lambda) = - \sum_{x,y} \tilde{P}(x,y) \log p(y|x; \lambda)$$

- 优化方法：
 - 梯度下降
 - 拟牛顿法
 - L-BFGS

CRF 解码-- Viterbi

$$p(Y|X) = \frac{1}{Z(X)} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, x_n) \right\}$$

- 韦特比变量 $\delta_t(i)$
- $\delta_t(i)$ 的含义是，给定模型 λ ，在时刻 t 处于状态 i ，观察到 $o_1 o_2 o_3 \dots o_t$ 的最佳状态转换序列为 $q_1 q_2 \dots q_t$ 的概率。

$$\delta_t = \prod_{i=1}^t \exp \left[\sum_k \lambda_k f_k(y_{i-1}, y_i, x) \right]$$

- 递推公式：

$$\delta_{t+1} = \delta_t \cdot \exp \left[\sum_k \lambda_k f_k(y_t, y_{t+1}, x) \right]$$

CRF 词性标注

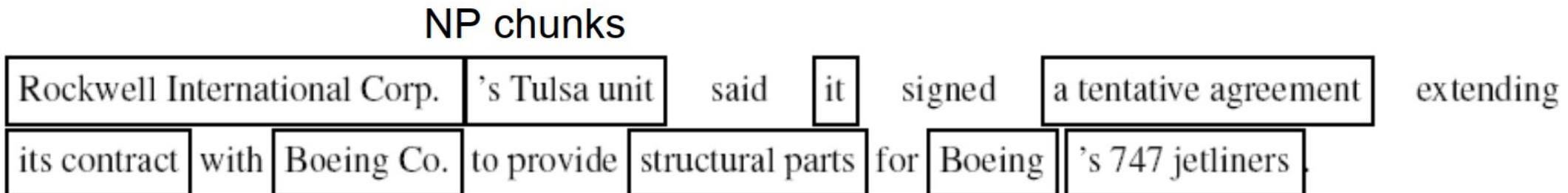
- w = The quick brown fox jumped over the lazy dog
- s = DET VERB ADJ NOUN-S VERB-P PREP DET ADJ NOUN-S
- Baseline is already 90%
 - Tag every word with its most frequent tag
 - Tag unknown words as nouns

Model	Error
HMM	5.69%
CRF	5.55%

Shallow Parsing

- 完整的parsing或信息提取的前身。
 - 识别文本中各种短语类型的非递归核心。
- 输入：带有POS tag单词的句子
- 任务：为每个单词打上标签，指示单词是否在短语块(chunk)之外(O)，是否开始一个短语块(B)，或者是否继续一个短语块(I)。
- CRF 在标准评估数据集上击败了所有单一模型的 NP 分块结果。

Model	F score
CRF	94.38%
Generalized winnow	93.89%
Voted perceptron	94.09%
MEMM	93.70%



Thank you!