

马尔可夫模型

隐马尔可夫模型 (HMM)

HMM的组成

HMM流程

前向算法

理论讲解

算法流程

后向算法

理论讲解

流程

Viterbi搜索算法

第一种解释

第二种解释

流程

参数学习

已知状态序列 (存在大量标注数据)

不存在大量标注数据

期望值最大化算法

前向后向算法

CRFs (没学懂)

马尔可夫模型

系统状态转移方程:

$$p(q_t = s_j \mid q_{t-1} = s_i, q_{t-2} = s_k, \dots)$$

- 假设1: 离散的二阶马尔可夫链 (当前状态只与前一个状态有关):

$$p(q_t = s_j \mid q_{t-1} = s_i, q_{t-2} = s_k, \dots) = p(q_t = s_j \mid q_{t-1} = s_i)$$

- 假设2: 在假设 1 基础上, 假设转态与时间无关

$$p(q_t = s_j \mid q_{t-1} = s_i) = a_{ij}, \quad 1 \leq i, j \leq N$$

该随机过程称为马尔可夫模型

马尔可夫模型中, 状态转移概率 a_{ij} 需要满足

$$a_{ij} \geq 0$$

$$\sum_{j=1}^N a_{ij} = 1$$

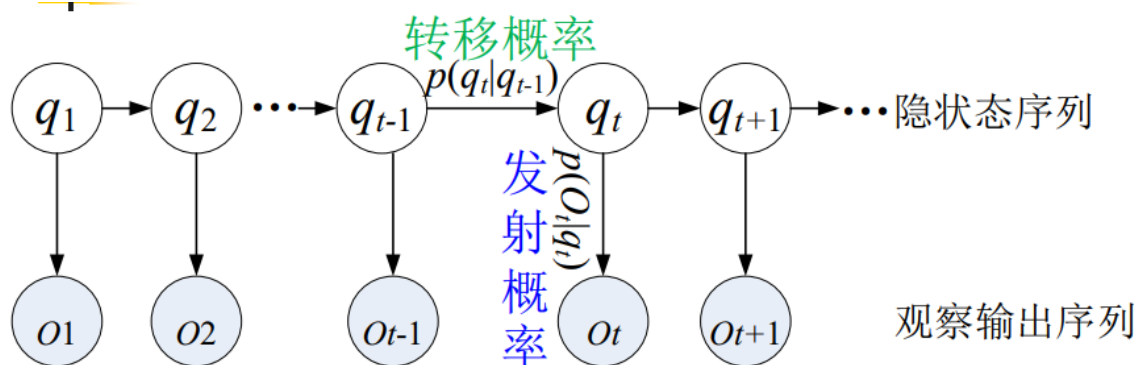
状态序列的概率为

$$\begin{aligned} p(s_1, \dots, s_T) &= p(s_1) \times p(s_2 | s_1) \times p(s_3 | s_1, s_2) \times \dots \times p(s_T | s_1, \dots, s_{T-1}) \\ &= p(s_1) \times p(s_2 | s_1) \times p(s_3 | s_2) \times \dots \times p(s_T | s_{T-1}) \\ &= \pi_{s_1} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \quad \dots (6.5) \end{aligned}$$

其中， $\pi_i = p(q_1 = s_i)$ ，为初始状态的概率。

隐马尔可夫模型 (HMM)

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



HMM的组成

- 模型中的转态数为 N
- 从每个状态可能输出的不同符号数 M
- 状态转移概率矩阵

$$\left\{ \begin{array}{l} a_{ij} = p(q_{t+1} = s_j | q_t = s_i), \quad 1 \leq i, j \leq N \\ a_{ij} \geq 0 \\ \sum_{j=1}^N a_{ij} = 1 \end{array} \right.$$

- 从一个状态中，观察到的符号的概率分布矩阵

$$\left\{ \begin{array}{l} b_j(k) = p(\overset{\text{状态的输出}}{o_t = v_k} | \underset{\text{状态}}{q_t = s_j}), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \\ b_j(k) \geq 0 \\ \sum_{k=1}^M b_j(k) = 1 \end{array} \right. \quad \dots (6.7)$$

- 初始状态的概率分布

$$\left\{ \begin{array}{l} \pi_i = p(\overset{\text{第一个状态是不同状态的概率}}{q_1 = s_i}), \quad 1 \leq i \leq N \\ \pi_i \geq 0 \\ \sum_{i=1}^N \pi_i = 1 \end{array} \right.$$

为了方便，一般将 HMM 记为： $\mu = (A, B, \pi)$
或者 $\mu = (S, O, A, B, \pi)$ 用以指出模型的参数集合。

HMM流程

给定模型 $\mu = (A, B, \pi)$, 产生观察序列 $O = o_1 o_2 \dots o_T$:

- (1) 令 $t=1$;
- (2) 根据**初始状态分布** $\pi = \pi_i$ **选择初始状态** $q_1 = s_i$;
- (3) 根据状态 s_i 的**输出概率分布** $b_i(k)$, **输出** $o_t = v_k$;
- (4) 根据**状态转移概率** a_{ij} , **转移到新状态** $q_{t+1} = s_j$;
- (5) $t = t+1$, 如果 $t < T$, 重复步骤 (3) (4), 否则结束。

前向算法

理论讲解

解决的问题: 已知输出序列和和给定模型, 快速计算观察序列的概率

给定模型 $\mu = (A, B, \pi)$ 和观察序列 $O = o_1 o_2 \dots o_T$, 快速计算 $p(O|\mu)$:

对于给定的状态序列 $Q = q_1 q_2 \dots q_T$, $p(O|\mu) = ?$

$$p(O|\mu) = \sum_Q p(O, Q|\mu) = \sum_Q \boxed{p(Q|\mu)} \times \boxed{p(O|Q, \mu)} \quad \dots (6.9)$$

$$p(Q|\mu) = \pi_{q_1} \times a_{q_1 q_2} \times a_{q_2 q_3} \times \dots \times a_{q_{T-1} q_T} \quad \dots (6.10)$$

$$p(O|Q, \mu) = b_{q_1}(o_1) \times b_{q_2}(o_2) \times \dots \times b_{q_T}(o_T) \quad \dots (6.11)$$

状态 q_1 生成观察值 o_1 的发射概率

状态 q_{T-1} 转换为状态 q_T 的转移概率

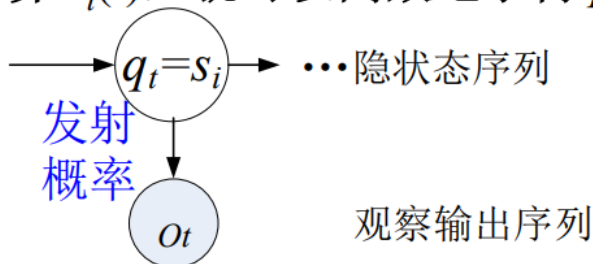
其实就是遍历所有可能产生的状态序列, 然后使用这些状态序列计算生成观察序列的概率, 然后求和

但是可能状态序列会很多, 难以直接搜索

- **解决办法：动态规划**
前向算法(The forward procedure)
- **基本思想**：定义前向变量 $\alpha_t(i)$ ：在时间 t ，输出序列 $o_1 o_2 \dots o_t$ 并且位于状态 s_i 的概率

$$\alpha_t(i) = p(o_1 o_2 \dots o_t, q_t = s_i | \mu) \quad \dots (6.12)$$

如果可以高效地计算 $\alpha_t(i)$ ，就可以高效地求得 $p(O|\mu)$ 。



利用前面变量计算概率。就是输出序列是固定的，对当前时刻所有状态求和

$$p(O | \mu) = \sum_{s_i} p(o_1 o_2 \dots o_T, q_T = s_i | \mu) = \sum_{i=1}^N \alpha_T(i)$$

$t+1$ 时间的前向变量可以根据 t 时刻前向变量递推得到

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

\uparrow 转移到 s_j 状态
 \downarrow 输出 O_{t+1}

t 时刻所有可能状态

算法流程

(1) 初始化: $\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N$

(2) 循环计算:

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

(3) 结束, 输出:

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

复杂度为 $O(N^2T)$

后向算法

理论讲解

定义后向变量 $\beta_t(i)$ 是在给定了模型 $\mu = (A, B, \pi)$ 和假定在时间 t 状态为 s_i 的条件下, 模型输出观察序列 $o_{t+1}o_{t+2} \cdots o_T$ 的概率:

$$\beta_t(i) = p(o_{t+1}o_{t+2} \cdots o_T \mid q_t = s_i, \mu) \quad \dots (6.15)$$

和前向算法一样, 用动态规划计算后向量

第一步

(1) 从时刻 t 到 $t+1$, 模型由状态 s_i 转移到状态 s_j , 并从 s_j 输出 o_{t+1} ;

概率为

$$a_{ij} \times b_j(o_{t+1})$$


第二步

(2) 在时间 $t+1$ ，状态为 s_j 的条件下，模型输出观察序列 $o_{t+2}o_{t+3}\cdots o_T$ 。

概率为 (以后面所有情况的和来计算当前时刻的后向量)

$$\beta_t(i) = \sum_{j=1}^N \overset{i \text{ 状态转到所有状态}}{a_{ij}} \underset{\text{输出}}{b_j(o_{t+1})} \times \overset{\text{后面的}}{\beta_{t+1}(j)}$$

归纳顺序为从后向前，因此称为后向算法

$$\beta_T(x), \beta_{T-1}(x), \cdots, \beta_1(x)$$


流程

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

(2) 循环计算:

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

(3) 输出结果: $p(O|\mu) = \sum_{i=1}^N \overset{\text{状态是 } s_i, \text{ 后面的全都生成了}}{\beta_1(i)} \times \overset{\text{第一个时刻生成 } O_1}{\pi_i \times b_i(o_1)}$

算法的时间复杂性: $O(N^2T)$

Viterbi搜索算法

问题: 如何发现最优状态序列，能够最好地解释观察序列

第一种解释

给定模型和输出序列，寻找每个时刻出现概率最大的状态

解释不是唯一的，关键在于如何理解“最优”的状态序列？一种解释是：状态序列中的每个状态都单独地具有概率，对于每个时刻 $t (1 \leq t \leq T)$ ，寻找 q_t 使得 $\gamma_t(i) = p(q_t = s_i | O, \mu)$ 最大。

我们可以将上面式子转化成和前向，后向算法相关的形式

$$\gamma_t(i) = p(q_t = s_i | O, \mu) = \frac{p(q_t = s_i, O | \mu)}{p(O | \mu)} \quad \dots (6.17)$$

模型的输出序列 O ，并且在时间 t 到达状态 s_i 的概率。

- 认为模型在时间 t 到达状态 s_i ，并且输出是 $O = o_1 \cdots o_T$
- 这可以拼接成前向（控制前面的输出）和后向变量（控制后面的输出）
 - (2) **根据前向变量的定义**（在时间 t ，输出序列 $o_1 o_2 \cdots o_t$ 并且位于状态 s_i 的概率），**实现这一步的概率为 $\alpha_t(i)$ 。**
 - (3) **根据后向变量的定义**（在时间 t 状态为 s_i 的条件下，模型输出观察序列 $o_{t+1} \cdots o_T$ 的概率），**实现这一步的概率为 $\beta_t(i)$ 。**
- 方程转化为

$$p(q_t = s_i, O | \mu) = \alpha_t(i) \times \beta_t(i)$$

分母以时间 t 的状态无关，下标 t 可以是任意的

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

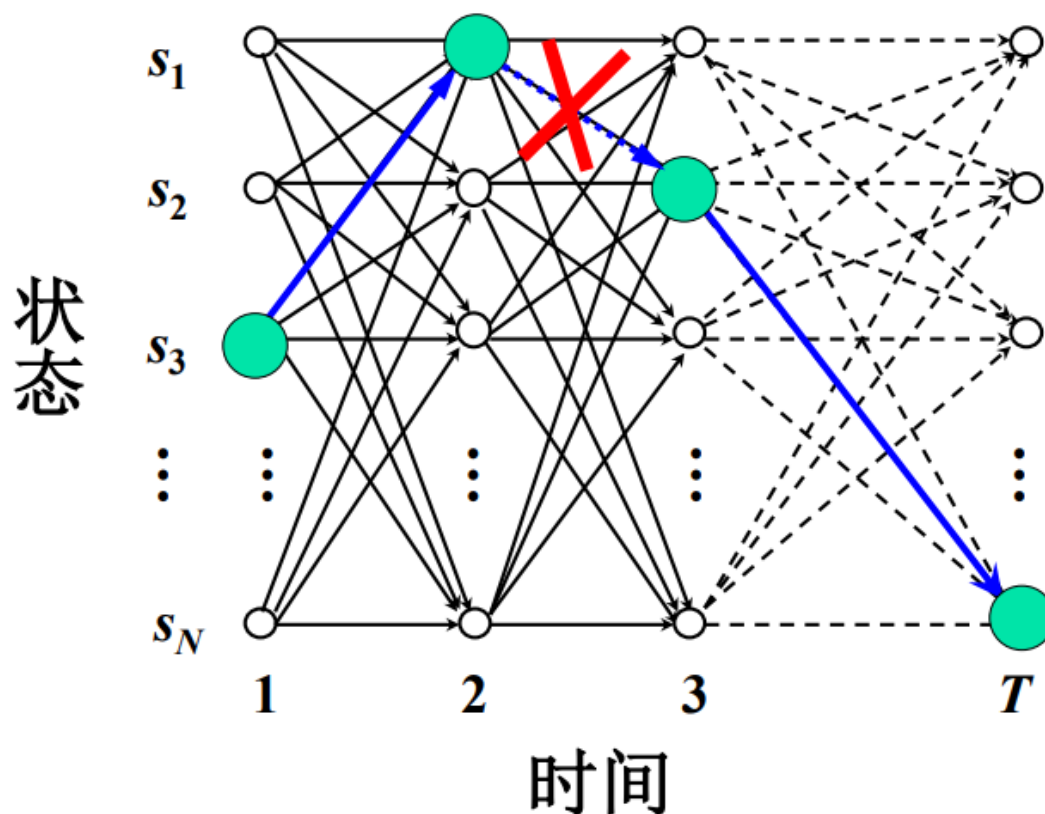
我们将分母与分子整合起来，得到

$$\gamma_t(i) = \frac{\alpha_t(i) \times \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \times \beta_t(i)}$$

因此，在 t 时刻，最优状态是

$$\hat{q}_t = \arg \max_{1 \leq i \leq N} (\gamma_t(i))$$

上面这种解释可能会有问题，因为每个状态单独最优不一定使整体的状态序列最优，可能两个最优状态之间的转移概率为 0



第二种解释

Viterbi算法：动态搜索最优状态序列

思想：从到前一个时间的所有最优路中选一个到当前时间的最优路

另一种解释：在给定模型 μ 和观察序列 O 的条件下求概率最大的状态序列：

$$\hat{Q} = \arg \max_Q p(Q | O, \mu) \quad \dots (6.21)$$

定义：Viterbi 变量 $\delta_t(i)$ 是在时间 t 时，模型沿着某一条路径到达 s_i ，并输出观察序列 $O = o_1 o_2 \dots o_t$ 的**最大概率**：

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p(q_1, q_2, \dots, q_t = s_i, o_1 o_2 \dots o_t | \mu) \quad \dots (6.22)$$

我们得到变量的递推公式

$$\delta_{t+1}(i) = \max_j [\delta_t(j) \cdot \overset{\text{从状态 } j \text{ 转到状态 } i}{a'_{ji}}] \cdot \overset{\text{状态 } i \text{ 输出 } O_{t+1}}{b_i(o_{t+1})}$$

解释：对于第 $t + 1$ 个时刻，会相对于 t 时刻增加一个节点，对于每一个状态，其前面的路径都是最优的。

- 类比于求最短路，当前节点为 v ， $\text{dis}[v] = \min(\text{dis}[u] + [u, v])$

• 流程

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

概率最大对应的路径变量： $\psi_1(i) = 0$

(2)递推计算：

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

走到节点 sj 的最优路径

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(o_t), \quad 2 \leq t \leq T, \quad 1 \leq i \leq N$$

记录当前状态 sj 是从前面哪条路走过来的 这一项可有可无，因为是对 i 求最优解

(3)结束:

$$\hat{Q}_T = \underset{1 \leq i \leq N}{\arg \max} [\delta_T(i)], \quad \hat{p}(\hat{Q}_T) = \max_{1 \leq i \leq N} \delta_T(i)$$

选择整体最优路的最后一个状态 左边是选路，这个是求概率

(4)通过回溯得到路径（状态序列）：

$$\hat{q}_t = \psi_{t+1}(\hat{q}_{t+1}), \quad t = T-1, T-2, \dots, 1$$

t+1时刻的状态 这个状态是从哪过来的

算法的时间复杂度： $O(N^2T)$

参数学习

刚刚都是给定模型和输出序列，求最优的状态序列

现在是给定一个观察序列，如何求得模型的参数，使得观察概率出现的概率最大

已知状态序列（存在大量标注数据）

用最大似然估计来计算参数：直接统计估计

$$\begin{aligned} \bar{\pi}_i &= \delta(q_1, s_i) \\ \bar{a}_{ij} &= \frac{Q \text{中从状态 } q_i \text{ 转移到 } q_j \text{ 的次数}}{Q \text{中所有从状态 } q_i \text{ 转移到另一状态(包括 } q_i \text{ 自身)的总数}} \\ &= \frac{\sum_{t=1}^{T-1} \delta(q_t, s_i) \times \delta(q_{t+1}, s_j)}{\sum_{t=1}^{T-1} \delta(q_t, s_i)} \quad \dots (6.24) \end{aligned}$$

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

$$\bar{b}_j(k) = \frac{Q \text{中从状态 } q_j \text{ 发射符号 } v_k \text{ 的次数}}{Q \text{ 到达 } q_j \text{ 的总次数}}$$

$$= \frac{\sum_{t=1}^T \delta(q_t, s_j) \times \delta(o_t, v_k)}{\sum_{t=1}^T \delta(q_t, s_j)} \quad \dots (6.25)$$

其中， v_k 是模型输出符号集中的第 k 个符号。

不存在大量标注数据

期望值最大化算法

- 初始化时随机地给模型的参数赋值（遵循约束的前提下，比如概率和为 1）
- 用当前模型，可以得到从某一状态转移到另一状态的期望次数。用这些期望次数得到新的模型参数
- 循环估计，参数收敛于最大似然估计

给定模型 μ 和观察序列 $O = o_1 o_2 \dots o_T$ ，那么，在时间 t 位于状态 s_i ，时间 $t+1$ 位于状态 s_j 的概率：

$$\xi_t(i, j) = p(q_t = s_i, q_{t+1} = s_j | O, \mu) = \frac{p(q_t = s_i, q_{t+1} = s_j, O | \mu)}{p(O | \mu)}$$

满足前面的输出及 t 时的状态
状态转移
满足后面的输出及 t+1 时的状态

$$= \frac{\alpha_t(i) \times a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}{p(O | \mu)}$$

$$= \frac{\alpha_t(i) \times a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \times a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)} \quad \dots (6.26)$$

那么，给定模型 μ 和观察序列 $O = o_1 o_2 \dots o_T$ ，在时间 t 位于状态 s_i 的概率为：

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad \dots (6.27)$$

对 j 求和，也就是从 i 转移到什么都可以

我们通过上面的值，利用下面的式子对模型参数进行重新估计

- 重新估计初始状态，转移概率和发射概率

(1) q_1 为 s_i 的概率:

$$\pi_i = \gamma_1(i)$$

(2) $\bar{a}_{ij} = \frac{\text{Q中从状态 } q_i \text{ 转移到 } q_j \text{ 的期望次数}}{\text{Q中所有从状态 } q_i \text{ 转移到下一状态(包括 } q_j \text{ 自身)的期望次数}}$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \dots (6.29)$$

(3) $\bar{b}_j(k) = \frac{\text{Q中从状态 } q_j \text{ 输出符号 } v_k \text{ 的期望次数}}{\text{Q到达 } q_j \text{ 的期望次数}}$

$$= \frac{\sum_{t=1}^T \gamma_t(j) \times \delta(o_t, v_k)}{\sum_{t=1}^T \gamma_t(j)} \quad \dots (6.30)$$

前向后向算法

就是上面的过程

- 参数随机初始化
- 执行 EM 算法

(2) 执行 EM 算法:

$$\xi_t(i, j) = \frac{\alpha_t(i) \times a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) \times a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)}$$

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

E-步: 由模型 μ_i 根据公式 (6.26) 和 (6.27) 计算期望值 $\xi_t(i, j)$ 和 $\gamma_t(i)$ 。

M-步: 用E-步中所得到的期望值，根据公式 (6.28-6.30) 重新估计 $\pi_i, a_{ij}, b_j(k)$ 得到模型 μ_{i+1} 。

循环: $i = i+1$ ，重复执行 E-步和M-步，直至 $\pi_i, a_{ij}, b_j(k)$ 的值收敛: $|\log p(O | \mu_{i+1}) - \log p(O | \mu_i)| < \varepsilon$ 。

- 结束

CRFs (没学懂)

基本思路：给定观察序列 X ，输出标识序列 Y 。通过计算 $P(Y|X)$ 计算最优标注序列

设 $G=(V, E)$ 为一个无向图， V 为结点集合， E 为无向边的集合， $Y = \{ Y_v | v \in V \}$ ，即 V 中每个结点对应于一个随机变量 Y_v ，其取值范围为可能的标记集合 $\{y\}$ 。如果以观察序列 X 为条件，每个随机变量 Y_v 都满足以下马尔可夫特性：

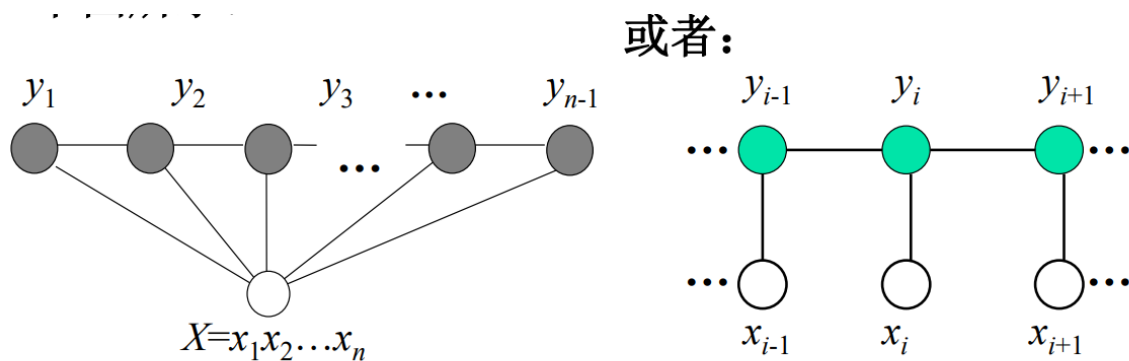
每个节点只受邻近节点影响

$$p(Y_v | X, Y_w, w \neq v) = p(Y_v | X, Y_w, w \sim v) \quad \dots (6-32)$$

其中， $w \sim v$ 表示两个结点在图中是邻近结点。那么， (X, Y) 为一个条件随机场。

为了增加标注的准确率，我们添加前一个词的标签，而不是只以当前位置的输入为依据

序列标注问题可以建模为简单的链式结构，在一定独立性限制的情况下， (X, Y) 也是条件随机场



在给定观察序列 X 时，某个特定标记序列 Y 的概率可以定义为：

$$\exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, X, i) + \sum_k \mu_k s_k(y_i, X, i)) \quad \dots (6-33)$$

其中， $t_j(y_{i-1}, y_i, X, i)$ 是转移函数，表示对于观察序列 X 的标注序列在 i 及 $i-1$ 位置上标记的转移概率；

$s_k(y_i, X, i)$ 是状态函数，表示观察序列 X 在 i 位置的标记概率；

λ_j 和 μ_k 分别是 t_j 和 s_k 的权重，需要从训练样本中估计出。