1 MLE 极大似然估计

1.1 似然函数

$$p(x|\theta)$$

如果 θ 是已知确定的，x 是变量，这个函数叫做概率函数 (probability function)，它描述对于不同的样本点 x，其出现概率是多少。

如果 x 是已知确定的，θ 是变量，这个函数叫做似然函数 (likelihood function), 它描述对于不同的模型参数，出现 x 这个样本点的概率是多少。

1.2 极大似然估计

现在对于一系列测试样本：

$$D = \{x_1, x_2, \cdots, x_N\}$$

在模型参数为 θ 时，每个 test 正确的概率是 $p(x_i|\theta)$, 所以极大似然函数是：

$$l(\theta) = p(D|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

为方便计算引入归一化对数似然函数

$$H(\theta) = \frac{1}{N} logp(D|\theta) = \frac{1}{N} \sum_{i=1}^{N} logp(x_i|\theta)$$

1.3 有趣的巧合

两个分布的 Cross-Entropy 为:

$$H(p|q) = H(p) + D_{KL}(p\|q)$$

KL-Divergence 离散形式定义为

$$D_{KL}(p\|q) = \sum_{y_i \in \Omega} p(y_i) log \frac{p(y_i)}{q(y_i)}$$

若结果 onehot 分布则

$$H(p|q) = \sum_{i=1}^{N} -log(p(y_i))$$

在 N 分类问题中，设 x 为 input case，y_i 为第 i 个类别，定义 Cross-Entropy 损失函数：

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} -log(p(y_i, \theta))$$

这与极大似然函数的形式恰好相同，这意味着离散状态下的 Minimize 交叉熵就是 Maximize 极大似然函数。

2 Markov 假设

所谓 Markov 假设，就是指 X_{t+1} 与 $X_{[0:t-1]}$ 关于 X_t 条件独立

$$P(X_{t+1}|\ X_{[0:t]}) = P(X_{t+1}|\ X_{[0:t-1]} \cap X_t) = P(X_{t+1}|\ X_t)$$

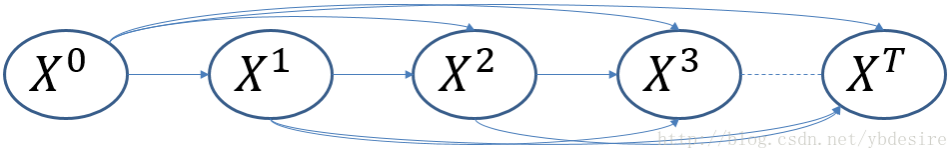
原来在一个贝叶斯网中, 某个长为 m 的序列出现的可能性为:

$$\begin{aligned} P(X_{[0:m]}) &= P(X_0, X_1, \cdots, X_m) \\ &= P(X_0)P(X_1|\ X_0)P(X_2|\ X_{[0:1]}) \cdots P(X_m|\ X_{[0:m-1]}) \\ &= \prod_{i=1}^{m} P(X_i|\ X_{[0:i-1]}) \end{aligned} \tag{1}$$

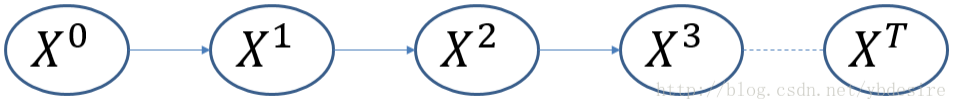
现在通过 Markov 假设，序列出现概率简化为:

$$P(X_{[0:m]}) = \prod_{t=0}^{m} P(X_{t+1}|X_t)$$

本质上是将一个贝叶斯网简化为 Markov Chain.



Bayesian Net



Markov Chain

2.1 语言模型中的应用

在大量文本中,一个长为 m 的单词序列 $[w_1, \cdots, w_m]$ 出现的概率只与 $[w_{1-n}, \cdots, w_1, \cdots w_m]$ 有关。这其实是对 Markov 假设的拓展，原始 Markov 假设的窗口长度为 1，而这里的窗口长度为 n.

$$p(w_1, w_2, \cdots, w_m) = \prod_{i=1}^{m} P(w_i|w_{[i-n:i]})$$

对于 Bigram 和 Trigram 有:

$$\begin{aligned} P(w_2|w_1) &= \frac{\#(w_1, w_2)}{\#(w_1)} \\ P(w_3|w_1, w_2) &= \frac{\#(w_1, w_2, w_3)}{\#(w_1, w_2)} \end{aligned} \tag{2}$$