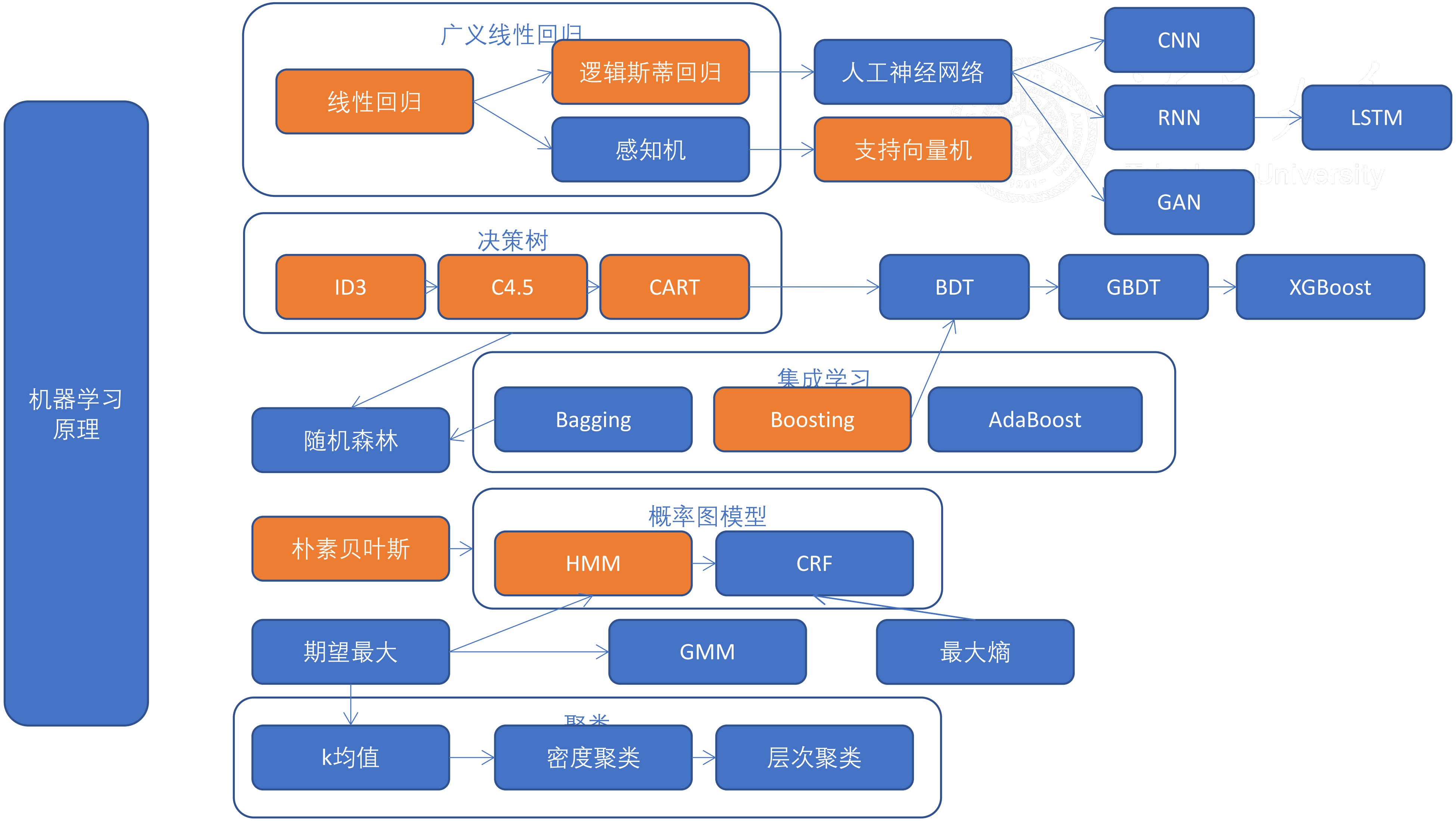


机器学习
原理
广义线性回归
线性回归
逻辑斯蒂回归
感知机
人工神经网络
支持向量机
CNN
RNN
LSTM
GAN
决策树
ID3
C4.5
CART
BDT
GBDT
XGBoost
随机森林
集成学习
Bagging
Boosting
AdaBoost
朴素贝叶斯
概率图模型
HMM
CRF
期望最大
GMM
最大熵
k均值
密度聚类
层次聚类

第十章
隐马尔科夫模型

隐马尔科夫模型的定义

- 隐马尔可夫模型是关于时序的概率模型;
- 描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列(state sequence)，再由各个状态生成一个观测而产生观测随机序列(observation sequence)的过程,序列的每一个位置又可以看作是一个时刻。

隐马尔科夫模型

- 组成
  - 初始概率分布
  - 状态转移概率分布
  - 观测概率分布
  - Q：所有可能状态的集合
  - V：所有可能观测的集合

$$Q=\{q_1,q_2,\cdots,q_N\}\,,\quad V=\{v_1,v_2,\cdots,v_M\}$$

  - I: 长度为T的状态序列
  - O：对应的观测序列

$$I=(i_1,i_2,\cdots,i_T)\,,\quad O=(o_1,o_2,\cdots,o_T)$$

隐马尔科夫模型

- 组成
  - A：状态转移概率矩阵

$$A=\left[a_{ij}\right]_{N\times N}$$

$$a_{ij}=P(i_{t+1}=q_j \mid i_t=q_i), \quad i=1,2,\cdots,N;\ j=1,2,\cdots,N$$

时刻 t 处于状态 q_i 的条件下在时刻 $t+1$ 转移到状态 q_j 的概率

隐马尔科夫模型

- 组成
  - B：观测概率矩阵

$$B=\left[b_j(k)\right]_{N\times M}$$

$$b_j(k)=P(o_t=v_k \mid i_t=q_j)\text{，}\quad k=1,2,\cdots,M;\ j=1,2,\cdots,N$$

在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率

  - π：初始状态概率向量

$$\pi_i=P(i_1=q_i)\text{，}\quad i=1,2,\cdots,N$$

时刻 $t=1$ 处于状态 q_i 的概率

隐马尔科夫模型

- 三要素

$$\lambda=(A,B,\pi)$$

- 两个基本假设
  - 齐次马尔科夫性假设，隐马尔可分链t的状态只和t-1状态有关：

$$P(i_t \mid i_{t-1},o_{t-1},\cdots,i_1,o_1)=P(i_t \mid i_{t-1})\,,\quad t=1,2,\cdots,T$$

  - 观测独立性假设，观测只和当前时刻状态有关；

$$P(o_t \mid i_T,o_T,i_{T-1},o_{T-1},\cdots,i_{t+1},o_{t+1},i_t,i_{t-1},o_{t-1},\cdots,i_1,o_1)=P(o_t \mid i_t)$$

例：盒子和球模型

• 盒子：	1	2	3	4
• 红球：	5	3	6	8
• 白球：	5	7	4	2

- 转移规则：
  - 盒子1 下一个 盒子2
  - 盒子2或3 下一个 0.4 左，0.6右
  - 盒子4 下一个 0.5 自身，0.5盒子3
- 重复5次：O={ 红，红，白，白，红}

例：盒子和球模型

- 状态集合：Q={盒子1， 盒子2， 盒子3， 盒子4}， N=4
- 观测集合：V={红球， 白球} M=2
- 初始化概率分布：

$$\pi = (0.25, 0.25, 0.25, 0.25)^{\mathrm{T}}$$

- 状态转移矩阵：

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

观测矩阵：

$$B = \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}$$

观测序列的生成过程

算法 10.1（观测序列的生成）

输入：隐马尔可夫模型 $\lambda=(A,B,\pi)$，观测序列长度 T；

输出：观测序列 $O=(o_1,o_2,\cdots,o_T)$．

（1）按照初始状态分布 π 产生状态 i_1

（2）令 $t=1$

（3）按照状态 i_t 的观测概率分布 $b_{i_t}(k)$ 生成 o_t

（4）按照状态 i_t 的状态转移概率分布 $\{a_{i_t i_{t+1}}\}$ 产生状态 i_{t+1}，　$i_{t+1}=1,2,\cdots,N$

（5）令 $t=t+1$；如果 $t<T$，转步 (3)；否则，终止

隐马尔科夫模型的三个基本问题

- 1、概率计算问题
- 给定： $\lambda=(A,B,\pi)$　$O=(o_1,o_2,\cdots,o_T)$
- 计算： $P(O|\lambda)$
- 2、学习问题
- 已知：$O=(o_1,o_2,\cdots,o_T)$
- 估计：$\lambda=(A,B,\pi)$，使 $P(O|\lambda)$ 最大
- 3、预测问题（解码）
- 已知：$\lambda=(A,B,\pi)$　$O=(o_1,o_2,\cdots,o_T)$
- 求：使 $P(I|O)$ 最大的状态序列 $I=(i_1,i_2,\cdots,i_T)$

前向算法

- 前向概率定义：给定隐马尔科夫模型λ，定义到时刻t部分观测序列为：$o_1, o_2, \cdots, o_t$，且状态为qi的概率为前向概率，记作：$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, i_t = q_i \mid \lambda)$$

算法 10.2（观测序列概率的前向算法）

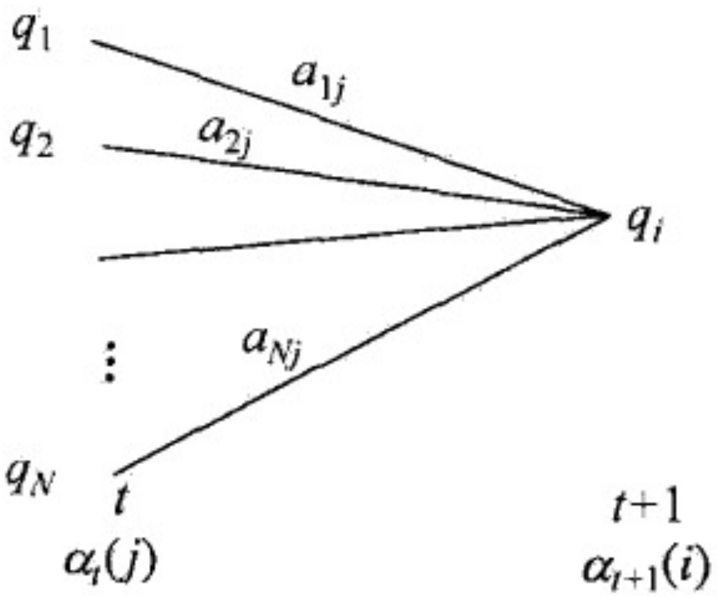
输入：隐马尔可夫模型 λ，观测序列 O；

输出：观测序列概率 $P(O \mid \lambda)$.

- 初值：$\alpha_1(i) = \pi_i b_i(o_1)$，　$i = 1, 2, \cdots, N$
- 递推：$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji} \right] b_i(o_{t+1}), \quad i = 1, 2, \cdots, N$$
- 终止：$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

前向算法

- 因为：$$\alpha_T(i)=P(o_1,o_2,\cdots,o_T,i_T=q_i\mid\lambda)$$
- 所以：$$P(O\mid\lambda)=\sum_{i=1}^{N}\alpha_T(i)$$
- 递推：



复杂度

$O(N^2T)$

前向算法

- 减少计算量的原因在于每一次计算，直接引用前一个时刻的计算结果，避免重复计算。

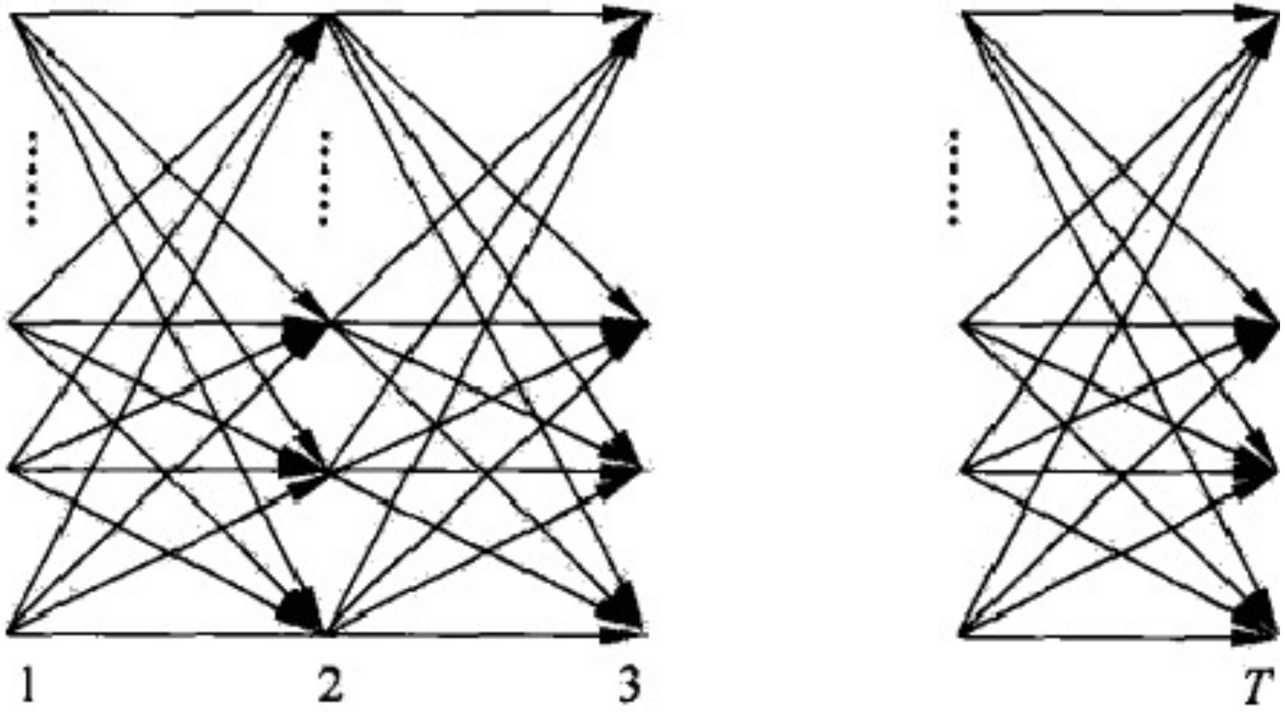


图 10.2　观测序列路径结构

复杂度

$O(N^2T)$

后向算法

- 定义10.3 后向概率：给定隐马尔科夫模型λ，定义在时刻t状态为qi的条件下，从t+1到T的部分观测序列为：$o_{t+1}, o_{t+2}, \cdots, o_T$ 的概率为后向概率，记作：

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T \mid i_t = q_i, \lambda)$$

可以用递推的方法求得后向概率 $\beta_t(i)$ 及观测序列概率 $P(O \mid \lambda)$

后向算法

算法 10.3 （观测序列概率的后向算法）

输入：隐马尔可夫模型 λ，观测序列 O；

输出：观测序列概率 $P(O\mid\lambda)$.

（1）

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

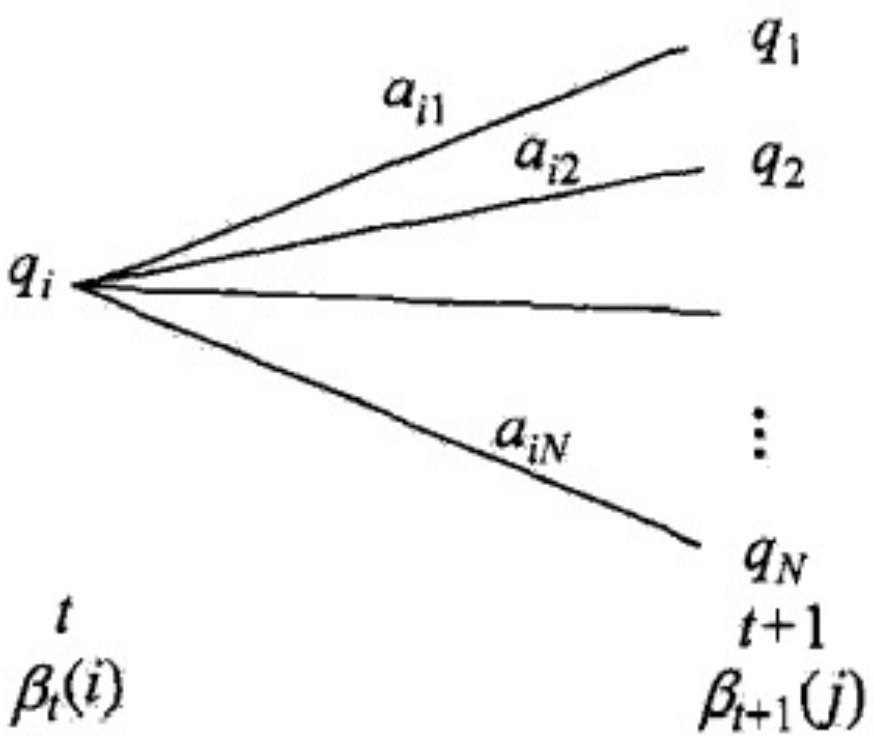
（2）对 $t=T-1,T-2,\cdots,1$

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(o_{t+1})\beta_{t+1}(j)\,,\quad i=1,2,\cdots,N$$

（3）

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

后向算法



- 前向后向统一写为：（t=1 和t=T-1分别对应）

$$P(O|\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)\,,\quad t=1,2,\cdots,T-1$$

一些概率和期望值的计算

1. 给定模型 λ 和观测 O，在时刻 t 处于状态 q_i 的概率.

记 $$\gamma_t(i) = P(i_t = q_i \mid O, \lambda)$$

$$\gamma_t(i) = P(i_t = q_i \mid O, \lambda) = \frac{P(i_t = q_i, O \mid \lambda)}{P(O \mid \lambda)}$$

$$\alpha_t(i)\beta_t(i) = P(i_t = q_i, O \mid \lambda)$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

一些概率和期望值的计算

2．给定模型 λ 和观测 O，在时刻 t 处于状态 q_i 且在时刻 $t+1$ 处于状态 q_j 的概率．记

$$\xi_t(i,j) = P(i_t = q_i, i_{t+1} = q_j \mid O, \lambda)$$

通过前向后向概率计算：

$$\xi_t(i,j) = \frac{P(i_t = q_i, i_{t+1} = q_j, O \mid \lambda)}{P(O \mid \lambda)} = \frac{P(i_t = q_i, i_{t+1} = q_j, O \mid \lambda)}{\sum_{i=1}^{N} \sum_{j=1}^{N} P(i_t = q_i, i_{t+1} = q_j, O \mid \lambda)}$$

$$P(i_t = q_i, i_{t+1} = q_j, O \mid \lambda) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

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

一些概率和期望值的计算

3．将 $\gamma_t(i)$ 和 $\xi_t(i,j)$ 对各个时刻 t 求和，可以得到一些有用的期望值：

（1）在观测 O 下状态 i 出现的期望值

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

（2）在观测 O 下由状态 i 转移的期望值

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

（3）在观测 O 下由状态 i 转移到状态 j 的期望值

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

Baum-Welch算法

- 假定训练数据只包括{O1,O2,...Os},
- 求模型参数λ=（A,B,π)
- 实质上是有隐变量的概率模型：EM算法

$$P(O|\lambda)=\sum_I P(O|I,\lambda)P(I|\lambda)$$

- 1、确定完全数据的对数似然函数
- 完全数据 $(O,I)=(o_1,o_2,\cdots,o_T,i_1,i_2,\cdots,i_T)$
- 完全数据的对数似然函数 $\log P(O,I|\lambda)$

Baum Welch算法

- 2、EM的E步 求 Q 函数 $Q(\lambda,\bar{\lambda})$

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

$$P(O,I\mid\lambda)=\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}b_{i_2}(o_2)\cdots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

- 则：

$$Q(\lambda,\bar{\lambda})=\sum_{I}\log\pi_{i_1}P(O,I\mid\bar{\lambda})$$
$$+\sum_{I}\left(\sum_{t=1}^{T-1}\log a_{i_t i_{t+1}}\right)P(O,I\mid\bar{\lambda})+\sum_{I}\left(\sum_{t=1}^{T}\log b_{i_t}(o_t)\right)P(O,I\mid\bar{\lambda})$$

- 对序列总长度T进行

Baum Welch算法

- 3、EM算法的M 步，极大化 $Q(\lambda,\bar{\lambda})$ 求模型参数A,B,π

第一项： $\sum_I \log \pi_{i_0} P(O,I|\bar{\lambda}) = \sum_{i=1}^{N} \log \pi_i P(O,i_1=i|\bar{\lambda})$

由约束条件： $\sum_{i=1}^{N} \pi_i = 1$ 利用拉格朗日乘子：

$$\sum_{i=1}^{N} \log \pi_i P(O,i_1=i|\bar{\lambda}) + \gamma\left(\sum_{i=1}^{N} \pi_i - 1\right)$$

求偏导数，并结果为0

- $$\frac{\partial}{\partial \pi_i}\left[\sum_{i=1}^{N} \log \pi_i P(O,i_1=i|\bar{\lambda}) + \gamma\left(\sum_{i=1}^{N} \pi_i - 1\right)\right] = 0$$
- 得： $P(O,i_1=i|\bar{\lambda}) + \gamma\pi_i = 0$ $\gamma = -P(O|\bar{\lambda})$ $\pi_i = \frac{P(O,i_1=i|\bar{\lambda})}{P(O|\bar{\lambda})}$

学习算法 Baum Welch算法

- 3、EM算法的M 步，极大化$Q(\lambda,\bar{\lambda})$ 求A,B,π

第二项可写成：

$$\sum_I\left(\sum_{t=1}^{T-1}\log a_{i_t i_{t+1}}\right)P(O,I\mid\bar{\lambda})=\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}\log a_{ij}P(O,i_t=i,i_{t+1}=j\mid\bar{\lambda})$$

由约束条件 $\sum_{j=1}^{N}a_{ij}=1$，拉格朗日乘子法：

- 得：

$$a_{ij}=\frac{\sum_{t=1}^{T-1}P(O,i_t=i,i_{t+1}=j\mid\bar{\lambda})}{\sum_{t=1}^{T-1}P(O,i_t=i\mid\bar{\lambda})}$$

Baum Welch算法

- 3、EM算法的M 步，极大化 $Q(\lambda,\bar{\lambda})$ 求A,B,π

第三项：

$$\sum_I\left(\sum_{t=1}^T \log b_{i_t}(o_t)\right)P(O,I\mid\bar{\lambda})=\sum_{j=1}^N\sum_{t=1}^T \log b_j(o_t)P(O,i_t=j\mid\bar{\lambda})$$

由约束条件： $\sum_{k=1}^M b_j(k)=1$

注意，只有在 $o_t=v_k$ 时 $b_j(o_t)$ 对 $b_j(k)$ 的偏导数才不为 0，

以 $I(o_t=v_k)$ 表示．求得

$$b_j(k)=\frac{\sum_{t=1}^T P(O,i_t=j\mid\bar{\lambda})I(o_t=v_k)}{\sum_{t=1}^T P(O,i_t=j\mid\bar{\lambda})}$$

学习算法 Baum Welch算法

- 将已上得到的概率分别用 $\gamma_t(i)$，$\xi_t(i,j)$ 表示：

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad b_j(k) = \frac{\sum_{t=1, o_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \pi_i = \gamma_1(i)$$

学习算法 Baum Welch算法

算法 10.4（Baum-Welch 算法）

输入：观测数据 $O=(o_1,o_2,\cdots,o_T)$；

输出：隐马尔可夫模型参数.

（1）初始化

对 $n=0$，选取 $a_{ij}^{(0)}$，$b_j(k)^{(0)}$，$\pi_i^{(0)}$，得到模型 $\lambda^{(0)}=(A^{(0)},B^{(0)},\pi^{(0)})$

（2）递推．对 $n=1,2,\cdots$，

$$a_{ij}^{(n+1)}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$b_j(k)^{(n+1)}=\frac{\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\pi_i^{(n+1)}=\gamma_1(i)$$

右端各值按观测 $O=(o_1,o_2,\cdots,o_T)$ 和模型 $\lambda^{(n)}=(A^{(n)},B^{(n)},\pi^{(n)})$ 计算

（3）终止．得到模型参数 $\lambda^{(n+1)}=(A^{(n+1)},B^{(n+1)},\pi^{(n+1)})$

预测算法

- 近似算法
- 想法：在每个时刻t选择在该时刻最有可能出现的状态 i_t^* ，从而得到一个状态序列 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$ 将它作为预测的结果，在时刻t处于状态qi的概率：

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{P(O\mid\lambda)}=\frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

- 在每一时刻t最有可能的状态是：$i_t^*=\arg\max_{1\leqslant i\leqslant N}[\gamma_t(i)]$，$t=1,2,\cdots,T$

从而得到状态序列：$I^*=(i_1^*,i_2^*,\cdots,i_T^*)$

得到的状态有可能实际不发生

维特比算法

- Viterbi 方法
- 用动态规划解概率最大路径，一个路径对应一个状态序列。
- 最优路径具有这样的特性：如果最优路径在时刻t通过结点 i_t^* ，那么这一路径从结点 i_t^* 到终点 i_T^* 的部分路径，对于从 i_t^* 到 i_T^* 的所有可能的部分路径来说，必须是最优的。
- 只需从时刻t=1开始，递推地计算在时刻t状态为i的各条部分路径的最大概率，直至得到时刻t=T状态为i的各条路径的最大概率，时刻t=T的最大概率即为最优路径的概率P*，最优路径的终结点 i_T^* 也同时得到。
- 之后，为了找出最优路径的各个结点，从终结点开始，由后向前逐步求得结点 $i_{T-1}^*, \cdots, i_1^*$，得到最优路径

$$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$$

维特比算法

- 导入两个变量δ和ψ，定义在时刻t状态为i的所有单个路径 $(i_1,i_2,\cdots,i_t)$中概率最大值为：

$$\delta_t(i)=\max_{i_1,i_2,\cdots,i_{t-1}} P(i_t=i,i_{t-1},\cdots,i_1,o_t,\cdots,o_1\mid\lambda),\quad i=1,2,\cdots,N$$

- 由定义可得变量δ的递推公式：

$$\begin{aligned}\delta_{t+1}(i)&=\max_{i_1,i_2,\cdots,i_t} P(i_{t+1}=i,i_t,\cdots,i_1,o_{t+1},\cdots,o_1\mid\lambda)\\&=\max_{1\leqslant j\leqslant N}[\delta_t(j)a_{ji}]b_i(o_{t+1}),\quad i=1,2,\cdots,N;\ t=1,2,\cdots,T-1\end{aligned}$$

- 定义在时刻t状态为i的所有单个路径中概率最大的路径的第t-1个结点为 $(i_1,i_2,\cdots,i_{t-1},i)$

$$\psi_t(i)=\arg\max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}],\quad i=1,2,\cdots,N$$

Viterbi 方法

算法 10.5（维特比算法）

输入：模型 $\lambda=(A,B,\pi)$ 和观测 $O=(o_1,o_2,\cdots,o_T)$；

输出：最优路径 $I^*=(i_1^*,i_2^*,\cdots,i_T^*)$.

（1）初始化

$$\delta_1(i)=\pi_i b_i(o_1), \qquad i=1,2,\cdots,N$$

$$\psi_1(i)=0, \qquad i=1,2,\cdots,N$$

（2）递推．对 $t=2,3,\cdots,T$

$$\delta_t(i)=\max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}]b_i(o_t), \qquad i=1,2,\cdots,N$$

$$\psi_t(i)=\arg\max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}], \qquad i=1,2,\cdots,N$$

Viterbi 方法

（3）终止

$$P^* = \max_{1 \leqslant i \leqslant N} \delta_T(i)$$

$$i_T^* = \arg\max_{1 \leqslant i \leqslant N} [\delta_T(i)]$$

（4）最优路径回溯．对 $t = T-1, T-2, \cdots, 1$

$$i_t^* = \psi_{t+1}(i_{t+1}^*)$$

求得最优路径 $I^* = (i_1^*, i_2^*, \cdots, i_T^*)$

• Q & A


七月在线
JULYEDU.COM
从零学AI的路线图：挑战年薪40万
通过七月在线《机器学习集训营》可解锁全部技能　超过2000人的成功经验：从零学AI、传统IT转型AI
Python基础
核心语法：循环 判断 控制
函数与面向对象
迭代器、生成器、装饰器
数据分析
科学计算之numpy
数据分析之pandas
数据分析实战（美国大选、房价预测）
数据结构
链表、队列、堆栈
字符串和数组
哈希表、树、图
查找与排序：增删改查
分治递归回溯
贪心和动态规划
概率与组合
大数据
Hadoop 基础(HDFS与YARN)
MapReduce与Hive SQL
分布式数据库Hbase
Spark与Flink
机器学习
机器学习的基本流程
经典模型：线性模型、决策树
常考模型：SVM与XGBoost
应用模型：HMM与CRF
特征工程：数据处理、模型构建/调优
上线部署：模型调参与模型评估
项目实战：图像检索、金融风控
深度学习
核心模型：CNN RNN LSTM
CV应用：Two-Stage和One-Stage框架
项目实战：调参、优化、模型压缩、蒸馏收敛
框架应用：TensorFlow与Pytorch
CV NLP 推荐的企业级项目实战
大规模跨境追踪/重识别（ReID）
人体关节点提取
智能客服系统
人体关节点提取
电商平台的商品推荐系统
从零开整电影推荐网站
简历指导、面试辅导、就业内推
CV 企业级项目实战
NLP 企业级项目实战
推荐的企业级项目实战
阶段一 夯实AI基础：数据分析与数据结构
阶段二 掌握AI核心：大数据/ML/DL
阶段三 CV NLP 推荐项目实战