对于EM算法的理解

背景

常见的思路是直接对极大似然函数MLE求梯度，令梯度等于0即为所求的最值（但其计算中会出现和的对数的形式，计算起来十分复杂），而**EM算法**就是为了解决“极大似然估计”这种更复杂而存在的。

Goal: Efficiently estimate the parameters of an HMM λ from an observation sequence X and known HMM topology $\mathcal{M}$: 找到能够最大化给定观察序列X的概率的参数λ

Parameters λ:

- Transition probabilities $a_{ij} = p(q_{t+1} = j \mid q_t = i)$
- Observation probabilities $b_j(x) = p(x|q = j)$

Maximum likelihood training: find the parameters that maximize

$$F_{ML}(\lambda) = \log p(X|\lambda) = \sum_{\boldsymbol{x}_i \in X} \log p(\boldsymbol{x}_i|\lambda) = \sum_{\boldsymbol{x}_i \in X} \log \sum_{Q \in Q} p(\boldsymbol{x}_i, Q|\lambda)$$

But likelihood doesn't factorize since observations not i.i.d.

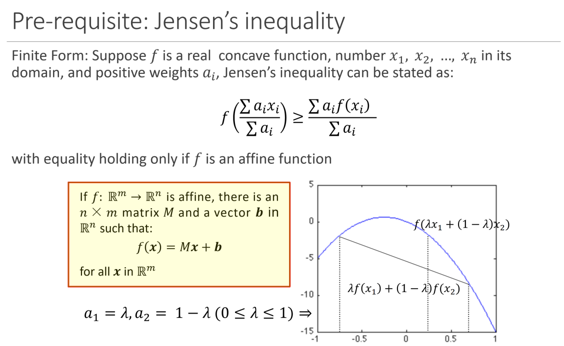
hidden variables – state sequence $Q = [q_1, \ldots, q_T]$

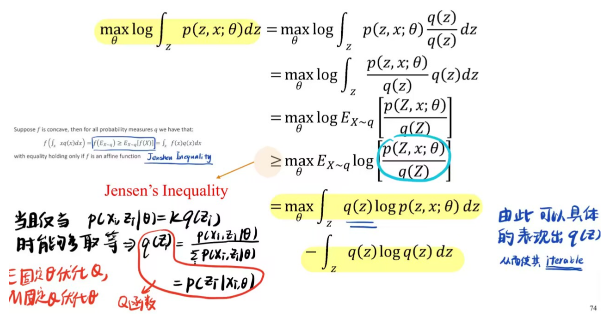
$$\boxed{\max_{\lambda} F(\lambda)}$$

由于HMM考虑了状态序列 $Q = [q_1, \ldots, q_T]$ 的影响，最大似然函数 $F_{ML}(\lambda)$ 不能简单地分解为独立观测的对数概率之和。相反，它是对所有可能的状态序列 Q 的概率分布 $p(x_i, Q|\lambda)$ 的对数求和。这意味着对于每个观测值 x_i，我们需要考虑所有可能的状态序列 Q 导致 x_i 的概率，并将这些概率的对数求和。

算法原理

Jensen不等式





根据取等条件，且由于$\sum q = 1$（权重和），故可知道$\sum_z p(x_i, z_j; \theta) = \frac{1}{k}$，此时，根据$p$和$q$之间的关系$p = kq$便有$q = \frac{\frac{1}{k}p(x_i,z_j;\theta)}{\frac{1}{k}} = \frac{p(x_i,z_j;\theta)}{\sum_z p(x_i,z_j;\theta)} = p(z_j|x_i;\theta)$

即可知，q就是后验概率。

EM (Expectation-Maximization)

1. 初始设置一个分布参数θ
2. 循环重复直到收敛
   - **E步，求Q函数**：对于每一个i，计算根据上一次迭代的模型参数来计算出隐性变量的后验概率（其实就是隐性变量的期望），来作为隐藏变量的现估计值（求在观测数据的前提下隐变量的期望）

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta)$$

   - **M步，求使Q函数获得极大时的参数取值**：将似然函数最大化以获得新的参数值（求经过隐变量改写的似然函数的极大）

$$\theta := \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}.$$

求θ的极值

$$\max_\theta \log \int_z p(z,x;\theta)dz \geq \max_\theta \int_z q(z)\log p(z,x;\theta)\,dz - \int_z q(z)\log q(z)\,dz$$

由于在M步骤中，$q(z)$被当作定值（不变），故优化时，只保留黄色的部分，其优化求解过程如下：

1. 求解$q(z)$，(8)式是贝叶斯公式，(9)是后验概率公式

$$P(X,Z;\theta) = P(Z;\pi)P(X|Z;\mu,\sigma) = \pi_k N(x_i;\mu_k,\sigma_k) \quad (8)$$

$$P(Z|X;\theta) = \frac{P(X,Z;\theta)}{\sum_Z P(X,Z;\theta)} = \frac{\pi_k N(x_i;\mu_k,\sigma_k)}{\sum_{k=1}^K \pi_k N(x_i;\mu_k,\sigma_k)} \quad (9)$$

2. 将$q(z)$代换，为最大化联合分布(μ^i,σ^i,π^i)，其中，π为初始状态，通过0梯度，拉格朗日法（有限制条件$\sum \pi_i = 1$，求$\frac{\partial L}{\partial \pi_i} = 0$）等方法求极值点

$$\begin{aligned}\theta^{(j+1)} &= \arg\max_\theta \sum_X \sum_Z Q^{(j)} log \frac{P(X,Z;\theta)}{Q^{(j)}} \\ &= \arg\max_\theta \sum_X \sum_Z (Q^{(j)} logP(X,Z;\theta) - Q^{(j)} logQ^{(j)}) \\ &= \arg\max_\theta \sum_X \sum_Z Q^{(j)} logP(X,Z;\theta)\end{aligned}$$

如此，不断缩小似然函数与其sup之间的距离，只要似然函数是有界的，只要M步中的0梯度点是极大值点，一直放大下去就能找到最终所求

Baum-Welch算法

HMM中的前向-后向算法

HMM的前向、后向概率估计:

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

$$P(O|\lambda)=\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)=\sum_{i=1}^{N}\alpha_T(i),1\le t\le T-1$$

可以用于计算：

- 在给定观测下每个时间点每个状态的概率（平滑概率）。

在时刻t处于状态q_i的概率

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

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

$$\xi_t(i,j)=P(i_t=q_i,i_{t+1}=q_j|O,\lambda)=\frac{P(i_t=q_i,i_{t+1}=q_j,O|\lambda)}{P(O|\lambda)}=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

- 状态转移的期望次数，这是参数估计（如Baum-Welch算法）中的一个关键步骤。

Baum-Welch的流程

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 $n=1,2,\cdots$，

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

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

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

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。式中 $\gamma_t(i)$，$\xi_t(i,j)$ 由式 (10.24) 和式 (10.26) 给出。

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

根据每一次的refinement得到最后模型的πAB，π表示初始值

参考资料

如何通俗理解EM算法-CSDN博客

文章浏览阅读10w+次，点赞1k次，收藏3.1k次。如何通俗理解EM算法前言 了解过EM算法的同学可能知道，EM算法是数据挖掘十大算法，可谓搞机器学习或数据挖掘的基本绕不开，但EM算法又像数据结构里的KMP算法，看似简单但又貌似不是一看就懂，想绕开却绕不开的又爱又恨，可能正在阅读此文的你感同身受。 一直以来，

https://blog.csdn.net/v_JULY_v/article/details/81708386?ops_request_misc={"request_id":"169969596616800222872034","scm":"20140713.130102334.."}&request_id=169969596616800222872034&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-81708386-null-null.142^v96^pc_search_result_base3&utm_term=EM算法&spm=1018.2226.3001.4187

https://vincentqin.gitee.io/blogresource-2/cv-books/statistical-learning-method.pdf