

最大熵模型

The Maximum Entropy Model

Xiachong Feng

说明：资料来源于：

1. IT 愚公的博客图解最大熵原理 (*The Maximum Entropy Principle*) ,
2. 《语言信息处理技术中的最大熵模型方法》
3. <http://www.52nlp.cn/maximum-entropy-model-tutorial-reading>
4. <https://blog.csdn.net/itplus/article/details/26550597>
5.

一 简单例子

1、2 来源于 IT 愚公的博客图解最大熵原理 (*The Maximum Entropy Principle*) , 3 来源于李素建《语言信息处理技术中的最大熵模型方法》

(1) 掷骰子问题

如果给你一颗骰子，你觉得分别掷到 1, 2, …, 6 的概率是多少？我觉得你肯定会立刻就回到我，答案是 $1/6$ 。很好，如果你知道答案是 $1/6$ ，就说明其实你已经掌握了最大熵原理。对于这个骰子问题，我们并不知道它到底是均匀的还是不均匀的，我们唯一知道的条件是 1-6 出现的总概率总是等于 1 的。我们把这个已知的条件称为“约束条件”。除了约束条件，我们对这个骰子一无所知，所以最符合现实的假设就是这个骰子是个均匀的骰子，每一点出现的概率是一个等概率事件。在这个问题中：

约束条件： $p(x = 1) + p(x = 2) + \dots + p(x = 6) = 1$

满足最大熵原理的概率： $p(x = 1) = p(x = 2) = \dots = p(x = 6) = \frac{1}{6}$

(2) 抽奖活动问题

假设你去参加抽奖活动，有 5 个盒子 ABCDE，奖品就放在这 5 个盒子中的一个，请问奖品在 ABCDE 盒子里的概率分别是多少？其实和骰子问题一样，我们除了知道奖品一定在其中一个盒子里（约束条件），但是对奖品到底在哪个盒子里一点额外的信息都没有，所以只能假设奖品在每个盒子里的概率都是 $1/5$ （等概率）。此时的约束条件是： $p(A) + p(B) + p(C) + p(D) + p(E) = 1$ ，概率是： $p(A) = p(B) = p(C) = p(D) = p(E) = \frac{1}{5}$ 。这时有个围观抽奖多时的吃瓜群众根据他的观察事实决定给你点提示，他说，奖品在 A 和 B 盒子里的概率总共是 $3/10$ 。此时的约束条件增加了，也就是说此时你知道了额外信息： $p(A) + p(B) = \frac{3}{10}$ ，你得到了新的约束条件。那么这个时候我再问你，奖品分别在各个盒子里的概率是多少呢？根据最大熵原理，我们加

入新的约束条件，然后继续把剩下的做等概率处理，虽然 $p(A) + p(B) = \frac{3}{10}$ ，但是我们并不知道 A 和 B 各自的贡献是多少，所以也只能假设它们两个贡献依然是平均的，所以各自就是 $3/20$ 。因为 $p(A) + p(B) = \frac{3}{10}$ ，所以 $p(C) + p(D) + p(E) = \frac{7}{10}$ ，但是我们依然不知道这三者的贡献，所以还是假设它们等概率，因此每一项就是 $7/30$ 。这就是通过最大熵原理处理概率问题了。

(3) 英汉翻译

以英汉翻译为例：对于英语中的“take”，它对应汉语的翻译有：

(t1)“抓住”：The mother takes her child by the hand. 母亲抓住孩子的手。

(t2)“拿走”：Take the book home. 把书拿回家。

(t3)“乘坐”：to take a bus to work. 乘坐公共汽车上班。

(t4)“量”：Take your temperature. 量一量你的体温。

(t5)“装”：The suitcase wouldn't take another thing. 这个衣箱不能装别的东西了。

(t6)“花费”：It takes a lot of money to buy a house. 买一所房子要花一大笔钱。

(t7)“理解、领会”：How do you take this package? 你怎么理解这段话？

假设对于所有的英文“take”，只有这七种翻译。则存在着如下限制： $p(t1|x) + p(t2|x) + p(t3|x) + \dots + p(t7|x) = 1$ ，表示在一个含有单词 take 的英文句子中，take 翻译为 t_i 的概率。在这个限制下，对每种翻译赋予均等一致的概率为： $p(t1|x) = p(t2|x) = p(t3|x) = \dots = p(t7|x) = \frac{1}{7}$ ，但是对于“take”，我们通过统计发现它的前两种翻译(t1)和(t2)是常见的，假设满足如下条件： $p(t1|x) + p(t2|x) = 2/5$ ，在两个限制条件下，分配给每个翻译的分布形式有很多，但是最一致的分布为：

$$p(t1|x) = p(t2|x) = \frac{1}{5}$$

$$p(t3|x) = p(t4|x) = p(t5|x) = p(t6|x) = p(t7|x) = \frac{3}{25}$$

可以验证，最一致的分布（满足限制条件的等概率分布）具有最大的熵。

但是上面的限制，都没有考虑上下文的环境，翻译效果不好。因此我们引入特征。例如，英文“take”翻译为“乘坐”的概率很小，但是当“take”后面跟一个交通工具的名词“bus”时，它翻译成“乘坐”的概率就变得非常大。为了表示 take 跟有“bus”时翻译成“乘坐”的事件，我们引入二值函数：

$$f(x, y) = \begin{cases} 1, & \text{if } y = \text{“乘坐” and } next(x) = \text{“bus”} \\ 0, & \text{otherwise} \end{cases}$$

x 表示上下文环境，这里看以看作是含有单词 take 的一个英文短语，而 y 代表输出，对应着“take”的中文翻译。 $next(x)$ 看作是上下文环境 x 的一个函数，表示 x 中跟在单

词 take 后的一个单词为“bus”。这样一个函数我们称作一个特征函数，或者简称一个特征。引入诸如上式中的特征，它们对概率分布模型加以限制，求在限制条件下具有最一致分布的模型，该模型熵值最大。

二 基本概念

(1) 最大熵原理

最大熵原理是统计学习的一般原理，将它应用到分类得到最大熵模型。最大熵原理认为，学习概率模型时，在所有可能的概率模型中，熵最大的模型是最好的模型。所以该准则可以表示为：在满足约束条件的模型集合中选取熵最大的模型。

(2) 熵的概念

熵 (entropy) 原本是热力学的一个概念，由香农引入到信息论中，在信息论和概率统计中，熵用来表示随机变量的不确定性。设 $X \in \{x_1, x_2, \dots, x_n\}$ 为离散型变量，其概率分布为， $p(X = x_i) = p_i, i = 1, 2, \dots, n$ ，则 X 的熵为：

$$H(X) = - \sum_{i=1}^n p_i \log p_i$$

当 $p_i = 0$ 时，定义 $0 \log 0 = 0$ 。由公式可知，熵依赖于 X 的分布，和具体值无关。 $H(X)$ 越大，表示不确定性越大。 $H(X)$ 的取值范围为 $0 \leq H(X) \leq \log n$ 。当对数分别以 2 和 e 为底的时候，熵的单位为比特 (bit) 和纳特 (nat)。在下面介绍拉格朗日乘子法之后，我们来证明这个不等式的成立问题。

下面介绍条件熵的概念，因为我们在分类模型之中用到的是条件概率。设 $X \in \{x_1, x_2, \dots, x_n\}$ ， $Y = \{y_1, y_2, \dots, y_m\}$ 为离散型变量，在已知 X 的条件下，Y 的条件熵 (conditional entropy) 可以定义为：

$$H(Y|X) = \sum_{i=1}^n p(x_i) H(Y|X = x_i) = - \sum_{i=1}^n p(x_i) \sum_{j=1}^m p(y_j|x_i) \log p(y_j|x_i)$$

(3) 拉格朗日乘子法及其对偶性原理

内容大部分来源于马同学高等数学-如何理解拉格朗日乘子法？

最大熵模型的推导需要用到拉格朗日乘子法，利用该方法，我们可以将带约束的极值问题转化为无约束极值问题进行求解。

形式化定义如下：求函数 $z = f(X)$ 在条件 $\phi_i(X) = 0 (i = 1, 2, 3, \dots, m)$ 下的可能极值点。其中 $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ ，那么具体的求解步骤如下：

- (i) 构造函数 $L(X) = f(X) + \sum_{i=1}^m \lambda_i \phi_i(X)$ ，其中 λ_i 为拉格朗日乘子。
- (ii) 求解方程组：

$$\begin{cases} \frac{\partial L}{\partial X} = 0 \\ \phi_i(X) = 0 (i = 1, 2, 3 \dots, m) \end{cases}$$

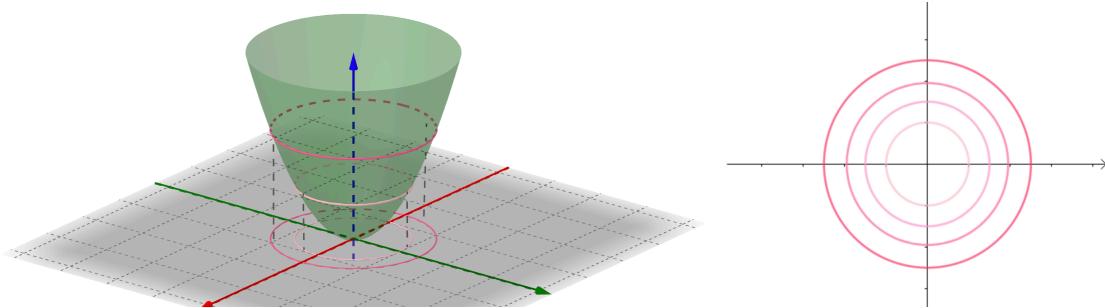
最终求解得到的 $x_1, x_2, \dots, x_n; \lambda_1, \lambda_2, \dots, \lambda_m$, 其中 x_1, x_2, \dots, x_n 就是函数可能的极值点。

下面来图像化的介绍一下拉格朗日乘子法, 因为该方法有很强的几何上的含义。

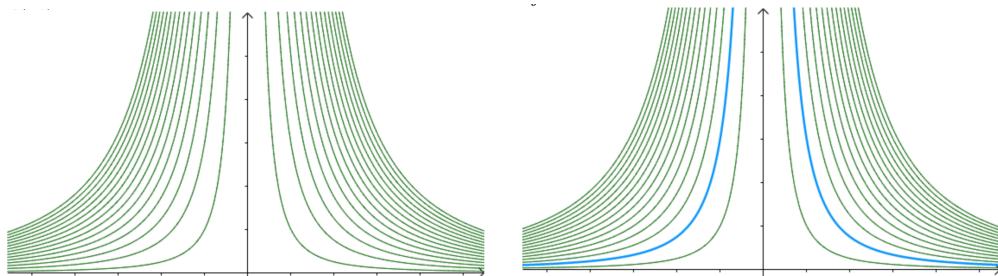
假设有自变量 x 和 y , 给定约束条件 $g(x, y) = c$, 要求 $f(x, y)$ 在约束条件 g 下的极值。设

$$\begin{aligned} f(x, y) &= x^2 + y^2 \\ g(x, y) &= x^2 y \end{aligned}$$

那么 $f(x, y)$ 以及其对应的等高线体现在图中为:

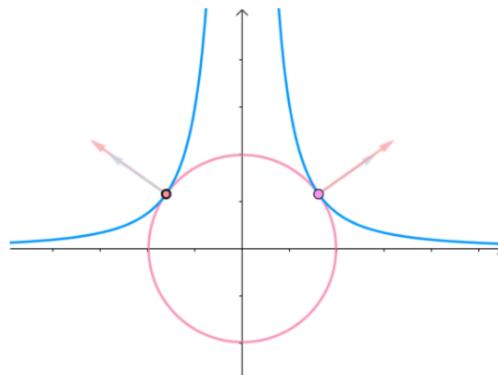


函数 $g(x, y)$ 的等高线以及其取一个特殊值为 3 的之后的等高线为:



此时我们可以将问题理解为, 根据 $f(x, y)$ 的等高线, 我们希望找到一个极值点, 使得 $f(x, y)$ 最小, 但是这个点同时需要满足约束条件 $x^2y = 3$, 也就是这个点需要在这个约束的线上 (上图蓝线) 上。

根据我们的直觉来讲, 当粉红色的圆与蓝色的相切的时候, 也就是满足下图所示的情况的时候, 相切的点既满足了极值点, 又满足了约束条件。

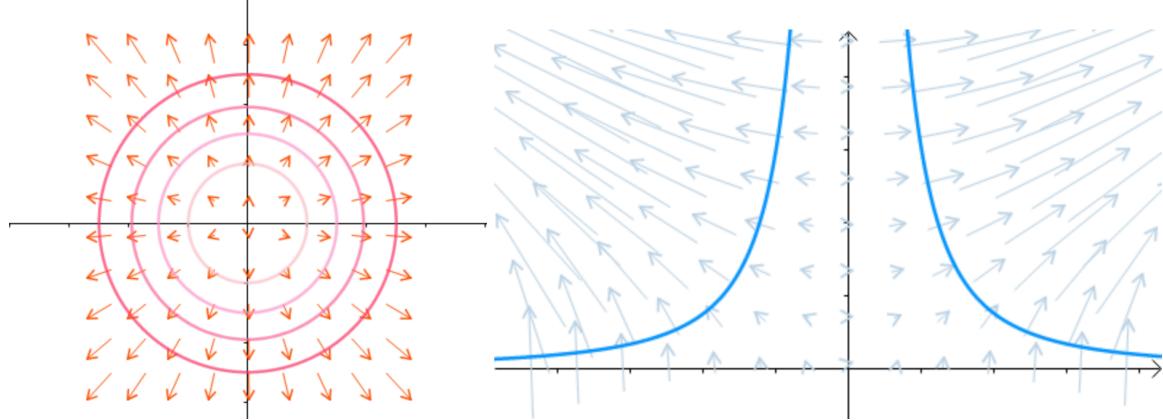


在相切的点，满足了两个限制条件：

$$\begin{cases} \text{在极值点，圆与曲线相切} \\ \text{梯度与等高线的切线垂直} \end{cases}$$

那么，在相切的点，圆的梯度向量和曲线的梯度向量平行。使用数学符号可以表述为：

$$\nabla f = -\lambda \nabla g$$



所以求解极值点 (x, y) ，我们联立方程组：

$$\begin{cases} \nabla f = -\lambda \nabla g \\ x^2 y = 3 \end{cases}$$

最后求解得到 x 、 y 、 $-\lambda$ 。将 $\nabla f = -\lambda \nabla g$ 进行变化得到 $\nabla(f(x, y) + \lambda g(x, y)) = \mathbf{0}$ 。可以根据此式理解上述拉格朗日乘子法求解过程。

下面简介拉格朗日对偶性 (Lagrange duality)。在最大熵模型的推导过程中也需要用到。我们往往通过将原始问题转换为对偶问题，通过求解对偶问题而得到原始问题的解。

首先介绍原始问题，假设 $f(x)$ ， $c_i(x)$ ， $h_j(x)$ 是定义在 R^n 上的连续可微函数。考虑约束最优化问题：

$$\begin{aligned} & \min_{x \in R^n} f(x) \\ & c_i(x) \leq 0, \quad i = 1, 2, 3, \dots, k \\ & h_j(x) = 0, \quad j = 1, 2, \dots, l \end{aligned}$$

称此约束最优化问题为原始最优化问题或者原始问题。为了求解该问题，引入拉格朗日函数， α_i 和 β_j 为拉格朗日乘子， $\alpha_i \geq 0$ 。

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^k \alpha_i c_i(x) + \sum_{j=1}^l \beta_j h_j(x)$$

考虑关于 x 的函数： $\theta_p(x) = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)$ ，注意这是一个关于 x 的函数，其中 $L(x, \alpha, \beta)$ 可以理解为 x 固定时，关于 α, β 的函数。

假设如果存在一个 x 不满足约束条件 $c_i(x) \geq 0$ 或者 $h_j(x) \neq 0$ 。那么 $\theta_p(x)$ 的max值为正无穷。如果满足约束条件则为 $f(x)$ 。

$$\theta_p(x) = \begin{cases} f(x), & x \text{ 满足约束条件} \\ \text{正无穷}, & \text{其他} \end{cases}$$

如果我们要求得就是在约束条件的极小值 (min)，因此等价于求：

$$\min_x \theta_p(x) = \min_x \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta)$$

这样原始问题转换为了极小极大问题。

(4) 小结

综合之前的三个例子，我们现在有一个直觉，系统中事件发生的概率满足一切已知约束条件，不对任何未知信息做假设，也就是对于未知的，当作等概率处理。用一句话来解释就是“model all that is known and assume nothing about that which is unknown”，在已知若干约束的情况下，我们建模时应该让模型满足这些约束，而对其他不作任何假设，在上面的例子中，不作任何假设就是“等概率”。

那么，如何将“最大熵”、“等概率”、“约束”等这些概念联系起来？熵实际上使用来衡量随机变量不确定性的，熵越大，说明系统越不稳定，越难猜测。我们希望除了已知的约束，我们对于未知模型的分布不参与干涉，不加入人为的先验知识。（例如你不可人为的猜测硬币材质不均匀，正反面出现的概率不一样）让所有可能出现的事情都有可能出现，体现为“等可能”。但是在上面的例子中，“等可能”是一件非常好理解的概念，但是系统越复杂，等可能的概念越来越难衡量，所以我们需要一个通用的衡量准则，这就是熵。越“等可能”，越不稳定，每部分越难猜测，“熵”越大。因此使用最大熵原理来选择模型可以（1）满足限制条件（2）不加入多余知识。

三 最大熵模型

给定一个训练数据集 $T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots (x_N, y_N)\}$ ，其中 $X \in \mathbb{R}^n$ 表示输入， Y 表示输出。如果我们想得到一个分类模型，我们需要计算条件概率 $P(Y|X)$ ，输入 X ，以条件概率输出 Y 。使用最大熵原理来选择最好的分类模型。

(1) 经验分布

根据数据集我们可以得到联合分布的经验分布和边缘分布的经验分布。经验分布式指通过训练数据进行统计得到的分布，如下：

$$\tilde{p}(x, y) = \frac{\text{count}(x, y)}{N}$$

$$\tilde{p}(x) = \frac{\text{count}(x)}{N}$$

(2) 特征函数

我们使用特征的概念来代表约束。特征使用特征函数来表示。特征函数 $f(x, y)$ 用来描述输入输出之间的一个事实，特征函数可以定义为任意实值函数，但是这里只讨论简单的二值定义。

$$f(x, y) = \begin{cases} 1, & \text{若 } x, y \text{ 满足某种事实} \\ 0, & \text{否则} \end{cases}$$

其实在例子部分已经涉及到了特征函数的概念，这里再举个例子来描述特征函数。

假设我们需要判断“打”字是动词还是量词，已知的训练数据有：

$$(x_1, y_1) = (\text{一打火柴}, \text{量词})$$

$$(x_2, y_2) = (\text{三打啤酒}, \text{量词})$$

$$(x_3, y_3) = (\text{五打塑料袋}, \text{量词})$$

$$(x_4, y_4) = (\text{打电话}, \text{动词})$$

$$(x_5, y_5) = (\text{打篮球}, \text{动词})$$

...

根据该简单数据集，可以提取两个特征函数：

$$f_1(x) = \begin{cases} 1, & \text{若“打”字前面为数字} \\ 0, & \text{否则} \end{cases}$$

$$f_2(x) = \begin{cases} 1, & \text{若“打”字后面为名词} \\ 0, & \text{否则} \end{cases}$$

(3) 约束条件

特征函数 $f(x, y)$ 关于经验分布 $\tilde{P}(x, y)$ 和真实分布 $P(x, y)$ 的期望分别是：

$$E_{\tilde{P}}(f) = \sum_{x,y} \tilde{P}(x, y) f(x, y)$$

$$E_P(f) = \sum_{x,y} P(x, y) f(x, y)$$

但是需要注意的是，和真实分布 $P(x, y)$ 我们是不可知的。同时我们的建模目标是 $p(y|x)$ ，使用贝叶斯定理进行转换之后，可以转换为：

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

我们期望仅仅使用训练数据集中的数据可以满足以下约束条件：

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

即我们希望特征的期望值应该和从训练数据中够得到的特征期望值是一致的。

(4) 最大熵模型

给定训练数据集，我们的目标是利用最大熵原理选择一个最好的分类模型，对于任意给定的输入 x ，可以以概率 $p(y|x)$ 输出 y 。

对于条件概率，我们有一个条件熵：

$$H(p(y|x)) = - \sum_{x,y} \tilde{p}(x)p(y|x) \log p(y|x)$$

根据最大熵原理，我们要求熵最大的那一个模型。给定一个训练数据集 $T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)\}$ ，以及特征函数 $f_i(x, y), i = 1, 2, \dots, n$ ，最大熵模型的学习等价于学习约束最优化问题。

$$\begin{aligned} \max_{p \in C} H(p) &= - \sum_{x,y} \tilde{p}(x)p(y|x) \log p(y|x) \\ &\quad \sum_y p(y|x) = 1 \end{aligned}$$

根据一般求优化问题的思想，我们转换为求最小值：

$$\begin{aligned} \min_{p \in C} -H(p) &= \sum_{x,y} \tilde{p}(x)p(y|x) \log p(y|x) \\ &\quad \sum_y p(y|x) = 1 \end{aligned}$$

四 模型求解

根据之前介绍的拉格朗日乘子法，我们将约束最优化原始问题转换为无约束最优化问题，是一个极小极大问题 ($\min \max$)，再通过对偶问题等价性，转换为求解上一步得到的极大极小问题 ($\max \min$)。

(1) 原始问题与对偶问题

首先引入拉格朗日乘子 $w_0, w_1, w_2, \dots, w_n$ ，将约束最优化问题转换为无约束最优化问题，定义拉格朗日函数 $L(P, w)$ ：

$$\begin{aligned} L(P, w) &\equiv -H(P) + w_0 \left(1 - \sum_y P(y|x) \right) + \sum_{i=1}^n w_i (E_{\tilde{P}}(f_i) - E_P(f_i)) \\ &= \sum_{x,y} \tilde{P}(x) P(y|x) \log P(y|x) + w_0 \left(1 - \sum_y P(y|x) \right) \\ &\quad + \sum_{i=1}^n w_i \left(\sum_{x,y} \tilde{P}(x, y) f_i(x, y) - \sum_{x,y} \tilde{P}(x) P(y|x) f_i(x, y) \right) \end{aligned}$$

得到最优化的原始问题：

$$\min_{p \in C} \max_w L(P, w)$$

通过对偶问题等价性，转换为对偶问题：

$$\max_w \min_{p \in C} L(P, w)$$

(2) 求解过程

在求解内层的极小问题时，可以导出最大熵模型的解具有指数形式，求解外层极大问题时，将会发现其与极大似然估计的等价性。

首先求解 $\min_{p \in C} L(P, w)$ ，是一个关于 w 的函数。记作： $\Psi(w) = \min_{p \in C} L(P, w) = L(P_w, w)$ ，将它的解记作 P_w （我认为 w 可以暂时理解为固定的）。 $L(P, w)$ 是一个关于 P 的函数。因为我们此时的目标是找到一个 p ，使得 $L(P, w)$ 最小。所以我们对 p 求导。如下：

$$\begin{aligned} \frac{\partial L(P, w)}{\partial P(y|x)} &= \sum_{x,y} \tilde{P}(x) (\log P(y|x) + 1) - \sum_y w_0 - \sum_{x,y} \left(\tilde{P}(x) \sum_{i=1}^n w_i f_i(x, y) \right) \\ &= \sum_{x,y} \tilde{P}(x) \left(\log P(y|x) + 1 - w_0 - \sum_{i=1}^n w_i f_i(x, y) \right) \end{aligned}$$

令偏导数为 0， $\tilde{p}(x) > 0$ 的情况下，解得：

$$P(y|x) = \exp \left(\sum_{i=1}^n w_i f_i(x, y) + w_0 - 1 \right) = \frac{\exp \left(\sum_{i=1}^n w_i f_i(x, y) \right)}{\exp(1 - w_0)}$$

根据条件 $\sum_y p(y|x) = 1$ 得：

$$\begin{aligned} P_w(y|x) &= \frac{1}{Z_w(x)} \exp \left(\sum_{i=1}^n w_i f_i(x, y) \right) \\ Z_w(x) &= \sum_y \exp \left(\sum_{i=1}^n w_i f_i(x, y) \right) \end{aligned}$$

其具体求解过程如下，这里 $w = \lambda$ ：

$$\begin{aligned} &\frac{\partial L(p, \lambda)}{\partial p(y|x)} \\ &= \sum_{x,y} \tilde{p}(x) (\log p(y|x) + 1) - \sum_y \lambda_0 - \sum_{i=1}^n \lambda_i \left(\sum_{x,y} \tilde{p}(x) f_i(x, y) \right) \\ &= \sum_{x,y} \tilde{p}(x) (\log p(y|x) + 1) - \sum_x \tilde{p}(x) \sum_y \lambda_0 - \sum_{x,y} \tilde{p}(x) \sum_{i=1}^n \lambda_i f_i(x, y) \quad (\text{利用 } \sum_x \tilde{p}(x) = 1) \\ &= \sum_{x,y} \tilde{p}(x) (\log p(y|x) + 1) - \sum_{x,y} \tilde{p}(x) \lambda_0 - \sum_{x,y} \tilde{p}(x) \sum_{i=1}^n \lambda_i f_i(x, y) \\ &= \sum_{x,y} \tilde{p}(x) \left(\log p(y|x) + 1 - \lambda_0 - \sum_{i=1}^n \lambda_i f_i(x, y) \right) \quad (\text{提出 } \sum_{x,y} \tilde{p}(x)) \end{aligned}$$

从而得到：

$$\log p(y|x) + 1 - \lambda_0 - \sum_{i=1}^n \lambda_i f_i(x, y) = 0,$$

进一步求解：

$$p(y|x) = e^{\lambda_0 - 1} \cdot e^{\sum_{i=1}^n \lambda_i f_i(x, y)},$$

带入约束条件 $\sum_y p(y|x) = 1$ 得：

$$\sum_y p(y|x) = e^{\lambda_0 - 1} \cdot \sum_y e^{\sum_{i=1}^n \lambda_i f_i(x, y)} = 1,$$

之后可得：

$$e^{\lambda_0 - 1} = \frac{1}{\sum_y e^{\sum_{i=1}^n \lambda_i f_i(x, y)}}.$$

最后可得：

$$p_\lambda = \frac{1}{Z_\lambda(x)} e^{\sum_{i=1}^n \lambda_i f_i(x, y)},$$

$$Z_\lambda(x) = \sum_y e^{\sum_{i=1}^n \lambda_i f_i(x, y)}$$

还是按照统计学习方法中的符号定义， $Z_w(x)$ 为规范化因子， $f_i(x, y)$ 是特征函数。 w_i 是特征的权值。最后得到的 $P_w(y|x)$ 就是最大熵模型，其中 w 是最大熵模型的参数向量。

之后求解问题外部的极大化问题 $\max_w \psi(w)$ ，我们目前已经知道了 p ，最后需要得到 w^* 。

$$w^* = \arg \max_w \psi(w)$$

统计学习方法书 85 页的例题可以帮助很快地理解这个求解过程。

五 极大似然估计

本节主要证明对偶函数的极大化等价于最大熵模型的极大似然估计。

已知训练数据集的经验概率分布 $\tilde{P}(X, Y)$ ，那么对应的条件概率分布， $P(Y|X)$ 的对数似然函数如下。这里以指数形式引入经验分布，应该是为了之后求解方便。

$$L_{\tilde{P}}(P_w) = \log \prod_{x,y} P(y|x)^{\tilde{P}(x,y)} = \sum_{x,y} \tilde{P}(x,y) \log P(y|x)$$

带入之前求解的最大熵模型时的解：

$$\begin{aligned}
L_{\tilde{P}}(P_w) &= \sum_{x,y} \tilde{P}(x,y) \log P_w(y|x) \\
&= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y} \tilde{P}(x,y) \log Z_w(x) \\
&= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_x \tilde{P}(x) \log Z_w(x)
\end{aligned}$$

对于对偶函数 $\Psi(w)$, 可得:

$$\begin{aligned}
\Psi(w) &= \sum_{x,y} \tilde{P}(x) P_w(y|x) \log P_w(y|x) \\
&\quad + \sum_{i=1}^n w_i \left(\sum_{x,y} \tilde{P}(x,y) f_i(x,y) - \sum_{x,y} \tilde{P}(x) P_w(y|x) f_i(x,y) \right) \\
&= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) + \sum_{x,y} \tilde{P}(x) P_w(y|x) \left(\log P_w(y|x) - \sum_{i=1}^n w_i f_i(x,y) \right) \\
&= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_{x,y} \tilde{P}(x) P_w(y|x) \log Z_w(x) \\
&= \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) - \sum_x \tilde{P}(x) \log Z_w(x)
\end{aligned}$$

可以发现这两个式子的结果是一样的。即 $\Psi(w) = L_{\tilde{P}}(P_w)$ 。

六 最优化算法

最大熵模型的学习归结为以似然函数为目标函数的最优化问题。常用的方法有迭代尺度法、梯度下降法、牛顿法或者拟牛顿法。

(1) 通用迭代尺度法 (GIS)

GIS 是一种通用的迭代算法, 其具体算法过程如下, 这里 $w = \lambda$:

Step 1 初始化参数. 令 $\lambda := 0$.

Step 2 计算 $E_{\bar{P}}(f_i)$, $i = 1, 2, \dots, n$.

Step 3 执行一次迭代, 对参数做一次刷新.

计算 $E_{p_\lambda}(f_i)$, $i = 1, 2, \dots, n$.

FOR $i = 1, 2, \dots, n$ DO

{

$\lambda_i := \lambda_i + \eta \log \frac{E_{\bar{P}}(f_i)}{E_{p_\lambda}(f_i)}$

}

Step 4 检查收敛条件, 若达到收敛条件则算法结束; 否则转至 Step 3.

其中, η 类似于学习率, 迭代公式中的:

$$\Delta \lambda_i = \eta \log \frac{E_{\bar{P}}(f_i)}{E_{p_\lambda}(f_i)},$$

为校正量。其中 \log 部分可以相当于梯度下降法中的梯度，但是本身并不是梯度。每一次迭代，先利用当前模型的参数来估算每个特征 f_i 在训练数据中的概率分布的期望 $E_{p_\lambda}(f_i)$ ，然后逐个与经验分布期望 $E_{\tilde{p}}(f_i)$ 作比较，偏差程度通过 $\log \frac{E_{\tilde{p}}(f_i)}{E_{p_\lambda}(f_i)}$ 进行刻画。

GIS 算法迭代时间很长，需要迭代很多次才可以收敛，不太稳定。

(2) 改进的迭代尺度法 (IIS)

我们的目标是通过极大似然估计学习模型参数，即求对数似然函数的极大值 \hat{w} 。IIS 的想法是如果可以有 $w + \delta$ 使得模型的对数似然函数比 w 大，那么就可以使用这种迭代，直到找到对数似然函数的最大值。其中 $w = (w_1, w_2, \dots, w_n)^T$ 。

改进之后的迭代算法大致流程如下所示：

Step 1 初始化参数. 令 $\lambda := 0$.

Step 2 执行一次迭代，对参数做一次刷新.

FOR $i = 1, 2, \dots, n$ DO

{

2.1 求解方程

$$\sum_{x,y} \tilde{p}(x)p(y|x)f_i(x,y)e^{\Delta\lambda_i \sum_{i=1}^n f_i(x,y)} = \tilde{p}(f_i)$$

得到 $\Delta\lambda_i$.

2.2 令 $\lambda_i := \lambda_i + \Delta\lambda_i$.

}

Step 3 检查收敛条件，若达到收敛条件则算法结束；否则转至 Step 2.

这里主要区别就在于如何计算 $\Delta\lambda_i$ ，主要通过求解一下方程得到。

$$\sum_{x,y} \tilde{p}(x)p(y|x)f_i(x,y)e^{\Delta\lambda_i \sum_{i=1}^n f_i(x,y)} = \tilde{p}(f_i)$$