

第6章 逻辑斯谛回归与最大熵模型

- ① 都是分类概率模型 $P(Y|X)$ 给定 X 要输出的
② 都是对数线性模型 概率分布

$$\ln P(Y|X) = w \cdot x$$

对概率取对数 在 logistic 是关于 x 的线性函数
变成关于 x, w 的 在最大熵模型里是关于 x 的函数的一个
线性函数 线性函数

③ 区别 logistic 判别模型

最大熵模型 生成模型 —

理解

(不仅 Y 有随机性
 X 也有随机性
 X 在样本里也对应一个
经验分布 $\hat{P}(X)$, 也就
是说在样本里给定 X
一个值, 就对应一个概率)

6.1 逻辑斯谛回归模型

公式的意义 **理解**

二项逻辑斯谛回归模型 要输入 X , 得到 $y \in \{0, 1\}$

$$\log \frac{P(Y=1|X)}{1-P(Y=1|X)} = w \cdot x \quad \text{是关于 } X \text{ 的函数, 记为 } \pi(x)$$

$$\Downarrow \quad P(Y=1|X) = \frac{\exp(w \cdot x)}{1 + \exp(w \cdot x)} \quad \text{即 } P(Y=1|X) = \pi(x)$$

但 logistic 是线性模型 即
 $w \cdot x$ (w 可以取 $(-\infty, +\infty)$). 但
 $P(Y=1|X) = \pi(x) \in [0, 1]$

模型参数估计: 极大似然估计

\because 要进行 logit 变化

极大似然估计涵义 (在概率模型上用得经常)

给出参数(待求解的)与样本的 $\underset{w}{\text{联合密度分布}}$ $\underset{\text{概率}}{\text{概率}}$

让概率最大, 求得 w . ★ (二项 logistic 回归模型整体思路)

$$P_w(y|x) = \pi(x)^y [1 - \pi(x)]^{1-y} \quad (\text{对任一个样本来说})$$

$$L(w) = \prod_{i=1}^N \pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i} \quad (\text{对 } N \text{ 个训练集来说})$$

即似然函数

$$\text{要求 } \max L(w)$$

$$\begin{aligned} \text{即 } \max \ln L(w) &= \sum_{i=1}^N \{ y_i (\ln \pi(x) + (1-y_i) \ln [1 - \pi(x)]) \} \\ &= \sum_{i=1}^N \{ y_i (w \cdot x_i) - \ln [1 + \exp(w \cdot x_i)] \} \end{aligned}$$

用梯度下降的方式求解

要求最大值, ∵ 要往正梯度更新 区别 感知机模型的
梯度下降模型
(往负方向更新,
∴ 要求最小值)

多项逻辑斯谛回归模型

Y的取值集合 $\{1, 2, \dots, K\}$ $X \in \mathbb{R}^{n+1}$, $w_k \in \mathbb{R}^{n+1}$

$$P(Y=k|X) = \frac{\exp(w_k \cdot X)}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot X)}, \quad k=1, 2, \dots, K-1$$

$$P(Y=K|X) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(w_k \cdot X)}$$

$$\ln \frac{P(Y=k|X)}{P(Y=K|X)} = w_k \cdot X \quad k=1, \dots, K-1$$

多项 logistic 中求了 $K-1$ 个 w_k

6.2 最大熵模型 (maximum entropy model)

最大熵原理:

在满足约束条件的模型集合中选择熵最大的模型

名词解释: $\tilde{P}(y_i)$ 表示在样本中 (观察到) 中出现 y_i 的概率

通过 谱本:

$$\min -H(P) = \sum_{i=1}^5 P(y_i) \log P(y_i)$$

$$\text{s.t. } P(y_1) + P(y_2) = \tilde{P}(y_1) + \tilde{P}(y_2) = \frac{3}{10}$$

$$\sum_{i=1}^5 P(y_i) = \sum_{i=1}^5 \tilde{P}(y_i) = 1$$

② 条件熵：(要给定一个 x , 在已知 x 的情况下, 求随机变量 y 的混乱程度)

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

对上式进行推导 (证)

一般: $H(p) = - \sum p_i \log p_i$

① $H(y|x) = - \sum_y P(y|x) \log P(y|x)$

② $E_x H(y|x)$ 注意这是小 x . 即一个取值

$$= - \sum_{xy} p(x) \cdot p(y|x) \log P(y|x) = H(p)$$

这是在给定 x 的条件下, 但 x 在样本中有很多取值, 如何综合 x 在不同取值下 y 的混乱程度呢? \Rightarrow
要求 $H(y|x)$ 的期望

$P(Y|X)$ 注意这个是大 X . 代表很多不同的 x 取值

∴ 条件熵公式由上可得出. 完毕

③ 最大熵模型就是要最大化条件熵 (将条件熵公式中★依然要利用一些已知的信息 $P(y|x) \Rightarrow \tilde{p}(y|x)$)

特征函数 $f(x,y) = \begin{cases} 1, & x \text{与 } y \text{ 满足某一事实} \\ 0, & \text{否则} \end{cases}$

观察到的一些信息.

$f_i(x, y)$ $i=1, \dots, N$ (表示观察到了 N 个事实)

我们要让这些事实在样本上出现的概率等于在整体上出现的概率，那我们如何用数学语言去描述这个事情呢？

\Rightarrow

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

↓
样本上

： f 是一个二级函数，要么为

0. 要么为 1. ∵ 在整体上的概率就 在整体上期望，可以用来表示

求 x, y 的联合概率分布

的期望

：回过头来看

最大熵模型的最优化问题

即最大化条件熵

$$\max_{P \in C} H(P) = - \sum_{x,y} \tilde{P}(x) P(y|x) \log P(y|x)$$

$$\text{s.t. } E_P(f_i) = E_{\tilde{P}}(f_i) \quad i=1, 2, \dots, n$$

$$\sum_y P(y|x) = 1$$

求得的结果

最大熵模型一般形式

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

其中 \downarrow 最大化似然函数

$$w = \arg \max_w L_{\tilde{P}} P(w) = \log \prod P(y|x) \tilde{P}(x, y)$$

模型参数

直观
意义理解

给定输入 x 分别求 y 取不同值的概率分布， f_i 前的求和表

示 n 个特征在给定的

x, y 满出现了几个， w_i 就是特征的重要程度。当我满足的特

征量越大，概率值就越

6.3 模型学习的最优化算法

①

改进的迭代尺度法

对数似然函数为

$$L(w) = \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n w_i \text{wifi}(x,y) - \sum_x \tilde{P}(x) \log Z_w(x)$$

来源：上一页中的 $\log \prod P(y|x) \tilde{P}(x,y)$

将 $P(y|x)$ 用 $\frac{1}{\sum w_i(x)} \exp(\sum w_i)$ 代替得

$Z_w(x)$ 为归一化系数，保证给定 x 情况下 y 的概率和为 1

② 拟牛顿法

$$\min_{w \in R^n} f(w) = \sum_x \tilde{P}(x) \log \sum_y \exp \left(\sum_{i=1}^n w_i \text{wifi}(x,y) \right) - \sum_{x,y} \tilde{P}(x,y) \cdot \sum_{i=1}^n w_i \text{wifi}(x,y)$$

理解

最大熵模型不同于之前的模型是用 x 来预测，

最大熵模型用的是 x 与 y 之间的特征关系

拉格朗日对偶性

用于解优化问题中的一个性质

附录 C.

① 优化问题 - P

一般形式 $\min f(x), x \in R^n$ (无约束最优化)

约束条件 \rightarrow s.t. $c_i(x) \leq 0, i=1, 2, \dots, k$
 $h_j(x) = 0, j=1, \dots, l$ 表示有 k 个不等式
约束

→ C 是不等式约束
L 是等式约束 拉格朗日乘子 所有的 c_i 都要 ≥ 0
 β 无限制

$$② L(x, \alpha, \beta) = f(x) + \sum_{i=1}^k \alpha_i c_i(x) + \sum_{j=1}^l \beta_j h_j(x)$$

↓ ↓
向量 向量
(k 个) (l 个)

目标函数 f
优化变量 x
可行域 } 在可行域内找到一个 x 使 $f(x)$ 最小
最优化的 x 记为 x^* $P^* = f(x^*)$

↓ 原问题等价于

$$P = \min_x \max_{\alpha, \beta} L(x, \alpha, \beta)$$

为什么呢 因为

$$P = \min_x \max_{\alpha, \beta} L(x, \alpha, \beta) = \begin{cases} f(x), & c_i(x) \leq 0, h_j(x) = 0 \\ \infty, & \text{其他} \end{cases}$$

∴ 有 $\boxed{\begin{array}{l} \min f(x) \\ \text{s.t. } c_i(x) \leq 0, h_j(x) = 0 \end{array}}$

② 原始问题

$$P \Leftrightarrow \min_{\alpha, \beta} \max_x L$$

拉格朗日对偶问题 \rightarrow 关于 α, β 求目标函数的 ^{最大} 值

$$\begin{aligned} & \max_{\alpha, \beta} \min_x L(x, \alpha, \beta) \\ & \text{st } d_i \geq 0 \end{aligned}$$

求得最优解为 α^*, β^*

定理 1 — 原始问题与对偶问题关系

d^* 与 P^* 关系

$$d^* = \max_{\alpha, \beta} \min_x L(x, \alpha, \beta) \leq \max_{\alpha, \beta} \min_{x \in \text{可行域}} L(x, \alpha, \beta) \leq \max_{\alpha, \beta} \min_x f(x)$$

即对偶问题是原始问题最优解的下界

原始问题提供了上界

$$PP \quad d^* \leq P^*$$

$$\begin{aligned} & \min f(x) \\ & = P^* \end{aligned}$$

定理 2. 什么时候 $d^* = P^* = L(x^*, \alpha^*, \beta^*)$

在原问题满足 2 个条件 等号成立

$$\left. \begin{array}{l} \text{① 原问题是 凸 优化问题} \\ \text{② Slater 条件} \end{array} \right\} \Rightarrow d^* = P^*$$

充分必要条件

强对偶性

名词解释

① 可行域 = 凸集 \rightarrow 凸优化问题

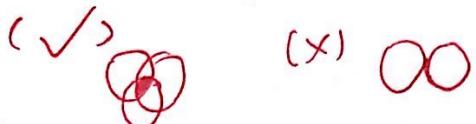
② $\min f(x)$: f 为凸函数

③ 凸集: 集合中任意 2 个点的连线还在集合内

④ 凸函数: 一元 \rightarrow 在函数上找 2 个点的连线在函数上方

(5) slater 条件：—— 较为宽松
针对约束条件中不等式约束，

各个约束条件可行域下有交集，并且交集中有点



意义

* 定理 2 告诉我们什么时候可以用拉格朗日对偶问题求解
优化问题

定理 3

提供另一个求解方式 — KKT — 只有原问题满足强对偶性
条件才可以

$$\left\{ \begin{array}{l} \nabla_x L(x^*, \alpha^*, \beta^*) = 0 \\ d_i^* c_i(x^*) = 0 \end{array} \right.$$

用

找到 x^* , α^* , β^* 满足
KKT 五个条件，即为解

$c_i(x^*) \leq 0$ — 原问题约束

$d_i^* \geq 0$ — 对偶问题约束

$b_i(x^*) = 0$ — 原问题约束

原问题 +
对偶问题

意义

若 $c_i(x^*) \neq 0$ 即 $d_i^* < 0$, x^* 没取到边界，即 $c_i(x)$ 没用

若 $c_i(x^*) = 0$, 则不对 d_i 作约束

6.3 改进的迭代尺度法

(对数似然函数)

$$L(w) = \sum_{x,y} \left[\tilde{P}(x,y) \sum_{i=1}^n w_i f_i(x,y) \right] - \sum_x \left[\tilde{P}(x) \ln Z_w(x) \right]$$

↓
 x, y 的经验分布
 ↓
 给出的特征
 变数
 取 0 or 1
 ↓
 给定 x 在训练集中
 的比例

直接求导不行(太复杂)

- ① 要给定一个 w 的初值
- ② 更新 w , 使 $L(w)$ 不断增大
- ③ 增大的量为

$Z_w(x)$ 是给定 x 的条件分布的归一化的系数.

$$= \sum_y \exp \left[\sum_i w_i f_i \right]$$

$$L(w+\delta) - L(w) = \sum_{x,y} \tilde{P}(x,y) \sum_{i=1}^n \delta_i f_i - \sum_x \tilde{P}(x) \ln \frac{Z_{w+\delta}(x)}{Z_w(x)}$$

要让增大的量尽可能大

但是 $\ln \frac{Z_{w+\delta}(x)}{Z_w(x)}$ 难处理 ∵ 利用 ~~核心~~ 核心

$$-\ln d \geq 1-d, d > 0$$

$$\Rightarrow \sum_x \tilde{P}(x) - \ln \frac{Z_{w+\delta}(x)}{Z_w(x)} \geq \sum_x \tilde{P}(x) \left[1 - \frac{Z_{w+\delta}(x)}{Z_w(x)} \right]$$

$$= 1 - \frac{\sum_x \frac{Z_{w+\delta}(x)}{Z_w(x)}}{\sum_x \tilde{P}(x)}$$

这样子就去掉了 \log

$$\frac{Z_{w+\delta}(x)}{Z_w(x)} = \frac{1}{Z_w(x)} \sum_y \exp \left(\sum_{i=1}^n (w_i + \delta_i) f_i \right)$$

$$\begin{aligned} \sum_y [P(y|x)] \exp(\sum \delta_i f_i) \\ = \sum_y \left[\frac{1}{Z_w(x)} \left(\exp \sum_{i=1}^n w_i f_i \right) \cdot \left(\exp \sum_{i=1}^n \delta_i f_i \right) \right] \end{aligned}$$

即

$$L(w+\delta) \geq \sum_i \tilde{p}(x,y) \sum_i \delta_i f_i + 1 - \sum_x \tilde{p}(x) \sum_y P_w(y|x)$$

$$- L(w) = \frac{\sum_y P_w(y|x) \exp(\sum_i \delta_i f_i)}{A(\delta|w)}$$

要找 δ 使 $L(w+\delta)$ 最大, \Leftrightarrow 要让 $A(\delta|w)$ 最大

但 δ_i 又出现在 指数位 (即下界最大)

\therefore 还要变化

要利用 Jensen 不等式

对一个凸函数 φ , 有权重 a_i
且 $\sum a_i = 1$

$$\text{就有 } \varphi\left(\sum a_i x_i\right) \leq \sum_i a_i \varphi(x_i)$$

$$\exp(\sum \delta_i f_i)$$

$$= \exp\left(\sum_i \frac{f_i}{f^*(x,y)} f^*(\delta_i)\right) \leq \sum_i \frac{f_i}{f^*} \exp(f^* f_i)$$

意义 把指数上的求和, 变成 先求指数再求和,

这样对 δ 求导时就不会涉及到其它分量.

即

$$L(w+\delta) \geq \sum_i \tilde{p}(x,y) \sum_i \delta_i f_i + 1 - \sum_x \tilde{p}(x) \sum_y P_w(y|x)$$

记作 $B(\delta|w)$

$$\sum_i \frac{f_i}{f^*} \exp(\delta_i f^*)$$

分别对每个 δ_i 求导, 使其 = 0

迭代尺度收敛的情况即 $L(w+\delta) = L(w)$. 差值为 0

代表 δ 为 0

$$\delta=0 \Rightarrow A(\delta|w)=0 \quad \text{— 第一次 放缩}$$

$$B(\delta|w)=0 \quad \text{— 第二次 放缩}$$

对每个 δ_i 求导有

$$g(\delta_i) = \sum_{x,y} \tilde{P}(x) P_w(y|x) f_i \exp(\delta_i f^*) - E_{\tilde{P}}(f_i)$$

要求零点 用牛顿法

要求 $g(\delta_i) = 0$. 用迭代

$$\delta_i^{(k+1)} = \delta_i^{(k)} - \frac{g(\delta_i^{(k)})}{g'(\delta_i^{(k)})}$$

