逻辑回归 (Logistic Regression)

李钦宾

先进智能计算与系统团队

邮箱：*qinbin@hust.edu.cn*

2024 年 3 月

目录

目 录

基本思想

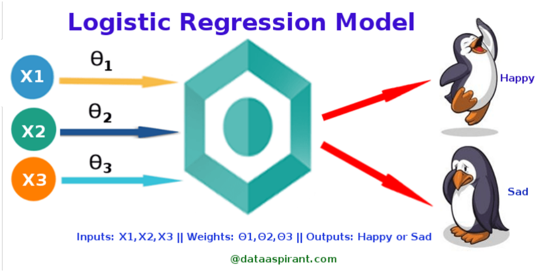


图: 逻辑回归模型分类示例

- 逻辑回归是用于分类任务的经典机器学习算法。
- 如图所示，将数据输入到模型中，然后模型给出它的分类结果。

基本思想

- 逻辑回归是高斯朴素贝叶斯 (连续特征的朴素贝叶斯) 的判别形式
- 机器学习算法可以大致分为两类:
  a) **生成算法**: 估计 $P(x_i, y)$(通常分别对 $P(x_i|y)$ 和 $P(y)$ 建模)
  b) **判别算法**: 判别模型 $P(y|x_i)$
- 朴素贝叶斯算法是生成算法。它对 $P(x_i|y)$ 建模，并对其分布 (例如，分类、多项、高斯⋯) 作了明确的假设。该分布的参数可以用 MLE 或 MAP 来估计。

回顾:
1) 贝叶斯公式:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

2) 贝叶斯分类器:

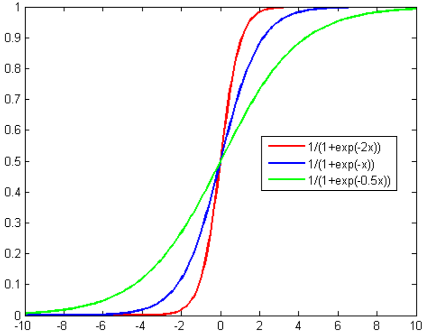
$$\underset{y}{\operatorname{argmax}}[p(y)\prod_{i=1}^{n} p(x_i|y,\theta)] = \underset{y}{\operatorname{argmax}}[log(p(y)) + \sum_{i=1}^{n} log(p(x_i|y,\theta))]$$

基本思想

- 上一讲曾给出，对于高斯朴素贝叶斯：

$$P(y|x_i) = \frac{1}{1 + e^{-y(w^T x_i + b)}}, y \in \{+1, -1\},$$

  其中向量 w 和 b 是通过 $P(x_i|y)$ 的选择来唯一确定的。
- 逻辑回归通常被称为朴素贝叶斯对应的判别式模型。这里我们为 $P(y|x_i)$ 建模，并假设它完全符合形式：$P(y|x_i) = \frac{1}{1+e^{-y(w^T x_i + b)}}$
- logistic(或 sigmoid 函数):



基本思想

- 我们可以对 $P(x_i|y)$ 作假设。例如，它可以是高斯分布或多项分布。但这并不重要，因为我们直接用 MLE 或 MAP 估计向量 w 和 b，以最大化 $\prod_i P(y_i|x_i; w, b)$ 的条件似然。

- 我们通过一个额外的常数维度 (类似于感知机) 将参数 b 吸收到 w 中。

$$P(y|X) = \frac{1}{1 + e^{-y(w^T X)}}$$

逻辑回归的 MLE

- 在 MLE 中，我们选择**最大化条件似然**的参数。条件似然 $P(Y|X,w)$ 是以训练数据中特征向量 x_i 为条件的观测值 $Y \in R^n$ 中的概率。其中，$X = [x_1, \ldots, x_i, \ldots, x_n] \in R^{d\times n}$。
- 我们选择使该函数最大化的参数，并假设对于给定的输入特征 x_i 和 w 时 y_i 之间是独立的。

$$P(\mathbf{y} \mid X, \mathbf{w}) = \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w}).$$

我们对上式取 log：

$$\log\left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \mathbf{w})\right) = -\sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

$$\hat{\mathbf{w}}_{MLE} = \underset{\mathbf{w}}{\operatorname{argmax}} -\sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

逻辑回归的 MLE

$$\hat{\mathbf{w}}_{MLE} = \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

- 我们需要估计参数 w。为了求出最小值对应的参数，可以试着求出 $\nabla_w \sum_i^n log(1 + e^{y_i w^T x_i}) = 0$。
- 该方程没有闭式解。我们在负对数似然上使用梯度下降来寻找近似解:

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$$

一维举例

考虑一个一维的情形，正类标记为加号，负类标记为圆。

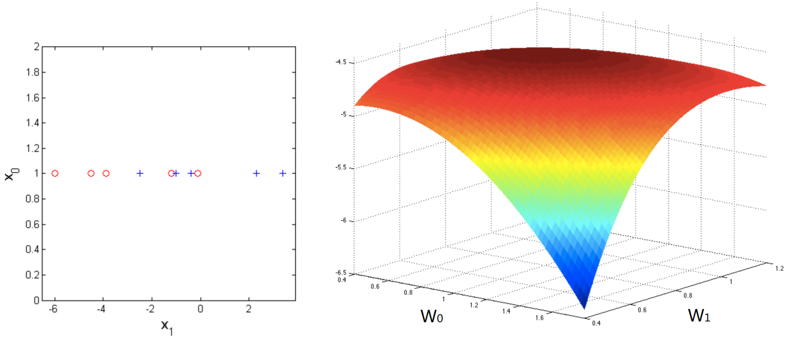


图: (a) 数据集：纵轴为常数特征 $x_0 = 1$　(b) 对数似然。

这个对数似然函数如右所示，在两个参数的空间中 (w_0, w_1)，最大值为 $(w_0 = 1, w_1 = 0.7)$。

一维情形

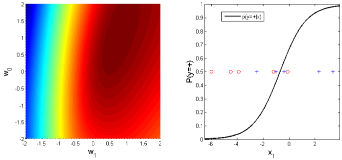
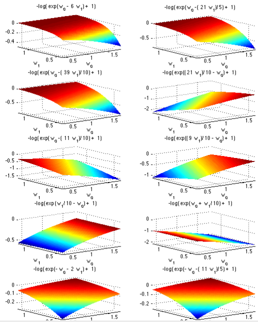


图: (a) 上页数据集的对数似然热图　(b) MLE 的解

下面是每个训练集中的样例对上述目标的贡献图:



二维情形

考虑一个二维的情形，正类标记为加号，负类标记为圆。

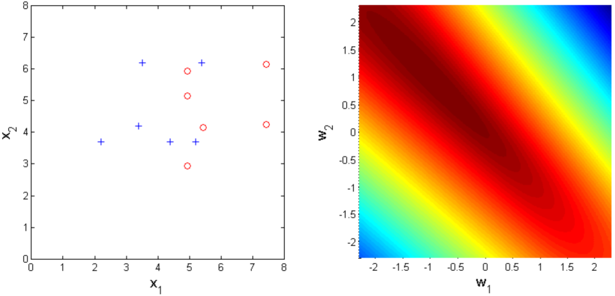


图: (a) 数据集　(b) Log 似然函数 (w_1, w_2) 在 $(-0.81, 0.81)$ 处取得最大值

二维情形

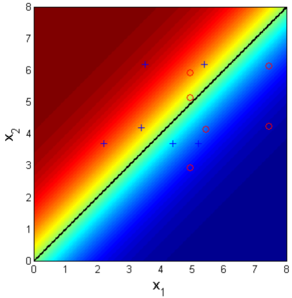


图: MLE 的解。其中红色表示正类的高概率。黑线表示 MLE 学到的决策边界

线性决策边界。为什么边界是线性的? 考虑边界处的条件:

$$P(Y=1|\mathbf{x},\mathbf{w}) = P(Y=-1|\mathbf{x},\mathbf{w}) \rightarrow \frac{1}{1+\exp\{-\mathbf{w}^\top\mathbf{x}\}} = \frac{\exp\{-\mathbf{w}^\top\mathbf{x}\}}{1+\exp\{-\mathbf{w}^\top\mathbf{x}\}} \rightarrow \mathbf{w}^\top\mathbf{x} = 0$$

对于上面的简单问题，最优 $w=(-0.81, 0.81)$，求解

$w^T x = 0.81 x_1 + 0.81 x_2 = 0 \qquad x_1 = x_2$

逻辑回归的 MAP

- 在 MAP 估计中，我们将 w 作为一个随机变量，并可以指定它的先验分布。假设先验为:　$w \sim N(0\ \sigma^2 I)$。这是逻辑回归的高斯近似。
- 我们在 MAP 中的目标是找到对给定数据的**最大化后验**的模型参数。

$$P(w|D) = \frac{P(D|w)P(w)}{P(D)} \propto P(D|w)P(w)$$

$$\begin{aligned} P(w|D) &= P(w|X,y) \propto P(X,y|w)P(w) \\ &= P(y|X,w)P(X)P(w) \propto P(y|X,w)P(w) \end{aligned}$$

$$\begin{aligned} \hat{w}_{MAP} &= \operatorname*{argmax}_{w} log(P(y|X,w)P(w)) \\ &= \operatorname*{argmin}_{w} \sum_{i=1}^{n} log(1+e^{y_i w^T x_i}) + \lambda w^T w \end{aligned}$$

$\lambda = \frac{1}{2\sigma^2}$。同样，这个函数没有闭式解，但我们可以在负对数后验上使用梯度下降来找到最优的参数。

$$\ell(\mathbf{w}) = \sum_{i=1}^{n} \log(1+e^{-y_i \mathbf{w}^T \mathbf{x}_i}) + \lambda \mathbf{w}^\top \mathbf{w}$$

小结

- 逻辑回归是朴素贝叶斯对应的判别模型。
- 在朴素贝叶斯中，我们首先为每个标签 y 建模 $P(x|y)$，然后获得最能区分这两个分布的决策边界。
- 在逻辑回归中，不对数据分布 $P(x|y)$ 建模，而是直接对 $P(y|x)$ 建模。
- 假设满足 $P(y|x_i) = \frac{1}{1+e^{y(w^T x_i + b)}}$，但并没有以任何方式限制对 $P(x|y)$ 的假设 (实际上它可以是指数函数)。
- 这使得逻辑回归更加灵活，但这种灵活性也需要更多的数据来避免出现过拟合现象。
- 通常，在数据很少且建模假设是合适的情况下，朴素贝叶斯一般要优于逻辑回归。然而，随着数据集变得越来越大，逻辑回归的表现往往优于朴素贝叶斯，这是因为对 $P(x|y)$ 所做的假设可能并不完全正确。
- 如果假设完全正确，即数据来自我们在朴素贝叶斯中假设的分布，那么逻辑回归和朴素贝叶斯在极限上可收敛到完全相同的结果 (但朴素贝叶斯会更快)。

朴素贝叶斯 vs. 逻辑回归

两种方法之间的区别:

Model	Naive Bayes	Logistic Regression
Assumption	$P(\mathbf{X}\|Y)$ is simple	$P(Y\|\mathbf{X})$ is simple
Likelihood	Joint	Conditional
Objective	$\sum_i \log P(y_i, \mathbf{x}_i)$	$\sum_i \log P(y_i \mid \mathbf{x}_i)$
Estimation	Closed Form	Iterative (gradient, etc)
Decision Boundary	Quadratic/Linear (see below)	Linear
When to use	Very little data vs parameters	Enough data vs parameters

目 录

基本思想

- $f(z) = h_\theta(x) = sigmoid(z) = \frac{1}{1+e^{-z}}$
- 输出 $=$ 0 或 1
- $Hypothesis \longrightarrow z = w^T x + b$

我们通常把 sigmoid 函数放在模型的最后，它将输出概率约束在 [0,1]。

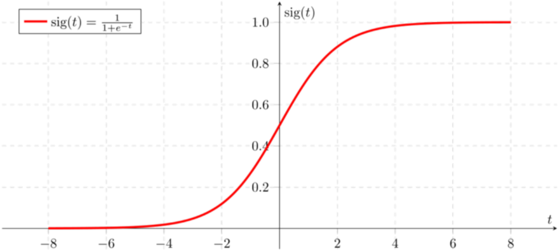


图: 如果 t→∞，y(预测) 将为 1，如果 t→-∞，y(预测) 将为 0。

损失函数

预测概率:

$$P(y=1|x) = h_\theta(x) = \frac{1}{1+e^{(-\theta^T x)}} = \sigma(\theta^T x) \tag{1}$$

$$P(y=0|x) = 1 - P(y=1|x) = 1 - h_\theta(x) \tag{2}$$

结合 (1) 和 (2):

$$P(y|x;\theta) = (h_\theta(x))^y (1-h_\theta(x))^{1-y} \tag{3}$$

y 表示标签，取值为 0 或 1

对给定的 m 个样本，使用最大似然估计 (MLE):

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m} (h_\theta(x^{(i)}))^{y^{(i)}} (1-h_\theta(x^{(i)}))^{1-y^{(i)}} \tag{6}$$

$$\longrightarrow \ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} (y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(y^{(i)}))) \tag{7}$$

$$J(\theta) = -\frac{1}{m}\ell(\theta)$$

$J(\theta)$ 是设计的损失函数

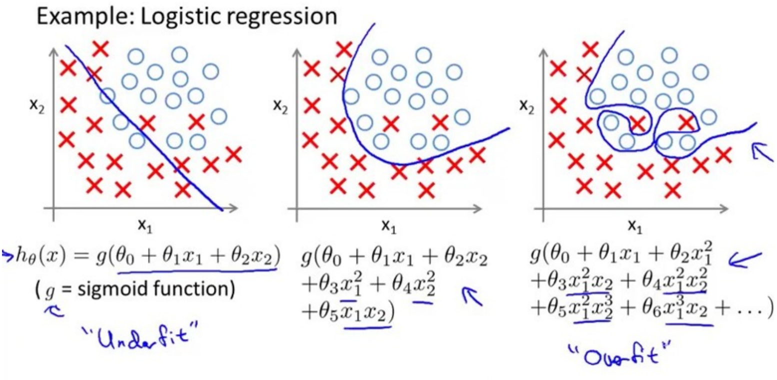
正则化 (Regularization)

基本思想

- L_2 正则：

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}))^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right] \tag{8}$$

- 正则化：防止权重变得过大
- 正则化可以通过对权值的约束在一定程度上避免过拟合



Example: Logistic regression
x_2
x_1
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
(g = sigmoid function)
"Underfit"
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$
"Overfit"

图: 欠拟合/过拟合

正则化

通过逻辑回归对任务进行分类

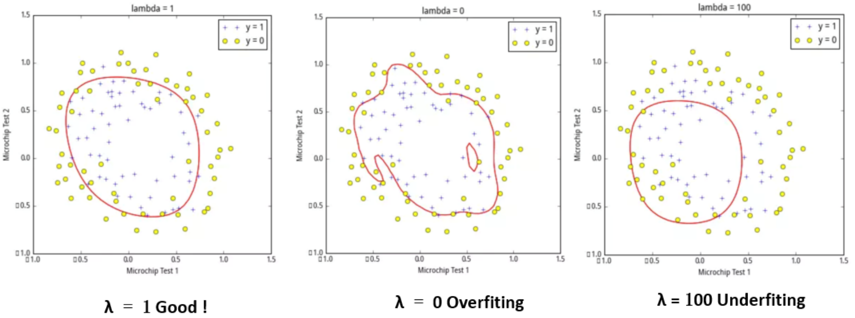
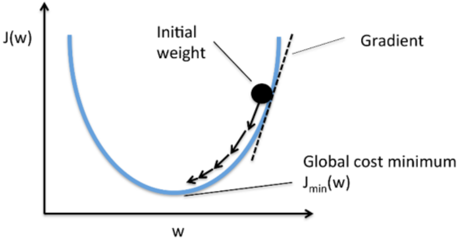
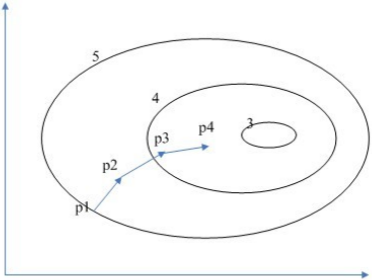


图: 不同 λ 值的二分类效果

梯度下降


J(w)
Initial
weight
Gradient
Global cost minimum
$J_{min}(w)$
w


5
4
3
p1
p2
p3
p4

梯度下降

$$J(\theta) = -\sum_{i=1}^{m}(y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(y^{(i)}))) \tag{9}$$

$$\downarrow$$

偏导数:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_i x_j^{(i)}(h_\theta(x^{(i)}) - y^{(i)})$$

$$\downarrow$$

权重更新:

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

逻辑回归 vs. 朴素贝叶斯

生成式 vs. 判别式

- 机器学习算法可以大致分为两类:
- 生成算法，估计 $P(x_i, y)$(通常分别对 $P(x_i|y)$ 和 $P(y)$ 建模)。
- 判别算法，使用模型 $P(y|x_i)$ 来生成 x_i 的预测输出.

1. 朴素贝叶斯算法是生成算法 ($p(y), p(x|y) \to p(y|x)$)
2. 逻辑回归算法是判别算法 (梯度下降 $\to w$)

$$P(y|x_i) = \frac{1}{1 + e^{-y(w^T x_i + b)}}$$

The End