> 本文主要是在阅读过程中对本书的一些概念摘录，包括一些个人的理解，主要是思想理解不涉及到复杂的公式推导。会不定期更新，若有不准确的地方，欢迎留言指正交流

第六章 逻辑斯特回归与最大熵模型

逻辑斯特回归（logistic regression）是统计学习中的经典分类方法。最大熵是概率模型学习的一个准则，将其推广到分类问题得到最大熵模型。**逻辑斯特回归模型与最大熵模型都属于 对数线性模型**

逻辑斯特回归模型

首先明确逻辑斯特分布函数的曲线，在中心附近增长速度较快，在两端增长速度较慢。其形状参数 γ 越小，曲线在中心附近增长越快。

X 具有下列分布函数和密度函数：

$$F(x)=P(X\leqslant x)=\frac{1}{1+\mathrm{e}^{-(x-\mu)/\gamma}} \tag{6.1}$$

$$f(x)=F'(x)=\frac{\mathrm{e}^{-(x-\mu)/\gamma}}{\gamma(1+\mathrm{e}^{-(x-\mu)/\gamma})^2} \tag{6.2}$$

式中，μ 为位置参数，$\gamma>0$ 为形状参数.

二项逻辑斯特回归模型是一种分类模型，由条件概率 P(Y|X) 表示，形式为参数化的逻辑斯特分布：

$$P(\mathrm{Y}=1\mid \mathrm{x})=\frac{\mathrm{e}^{w\bullet x+b}}{1+e^{w\bullet x+b}}$$

$$P\left(Y=0\mid x\right)=\frac{1}{1+e^{w\bullet x+b}}$$

其中 w 为权值向量，b 为偏置。 上述就是概率回归模型。

这里给出几率（odd）的概念，不同于概率，

> 几率（odd）是指一件事情发生的概率与该事件不发生的概率的比值

如果一件事情发生的概率为 p ，则该事件的对数几率就是：

logit(p) = log(p / (1-p))

对于逻辑斯特回归而言，上述对数公式就是：

$$\log\frac{P(Y=1\mid x)}{1-P(Y=1\mid x)}=w\bullet x$$

由上面公式可以得到，输出 Y = 1 的对数几率是由输入x 的线性函数表示的模型。

模型参数估计

对数似然函数为

$$L(w)=\sum_{i=1}^{N}[y_i\log\pi(x_i)+(1-y_i)\log(1-\pi(x_i))]$$
$$=\sum_{i=1}^{N}\left[y_i\log\frac{\pi(x_i)}{1-\pi(x_i)}+\log(1-\pi(x_i))\right]$$
$$=\sum_{i=1}^{N}[y_i(w\cdot x_i)-\log(1+\exp(w\cdot x_i)]$$

对 $L(w)$ 求极大值，得到 w 的估计值.

由上问题就变成对数似然函数为目标函数的优化问题，逻辑斯特回归学习中通常使用**梯度下降法**及**拟牛顿法**。

上面的二项逻辑斯特回归可以同等的推广到多项逻辑斯特回归。

$$P(Y=k|x)=\frac{\exp(w_k\cdot x)}{1+\sum_{k=1}^{K-1}\exp(w_k\cdot x)},\quad k=1,2,\cdots,K-1 \qquad (6.7)$$

$$P(Y=K|x)=\frac{1}{1+\sum_{k=1}^{K-1}\exp(w_k\cdot x)} \qquad (6.8)$$

最大熵模型

> 最大熵原理：在学习概率模型时，在所有可能的概率模型分布中，熵最大的模型就是最好的模型

所以用最大熵原理作为约束条件选取的模型就是最大熵模型。最大熵原理通过熵的最大化来表示等可能性。

所以在模型学习的约束条件就是，特征函数对于联合分布P（X，Y）的经验分布P'（X，Y）的期望与特征函数关于模型 P（Y|X）与经验分布P'（X）的期望值相等。

满足上诉约束条件的模型集合 C 中求条件熵 H(P)最大的模型。（关于条件熵的定义及定义式见：5.决策树）

最大熵模型学习

因为有条件熵的定义式H（P），有约束条件，所以这里引入拉格朗日乘子 w，定义一个拉格朗日函数 L（P，w），这里 L（P，w）是关于 P 的凸函数，然后对拉格朗日函数进行求解，先求解拉格朗日的最小化，然后得到关于 w 的表达式，再对这个表达式求最大化，得到 w 代入最终的结果。

第二步对 w 的求解过程中，是对偶函数的极大化，其就等同于最大熵模型的极大似然估计。

模型的最优化方法

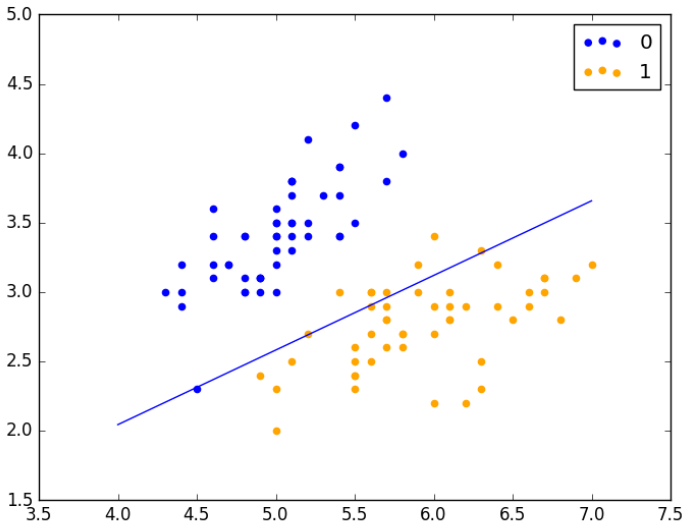
逻辑斯特回归模型和最大熵模型都是对数线性模型，模型学习就是再给定的训练数据条件下对模型进行极大似然估计或正则化的极大似然估计。

而模型的最优化算法常用的有**改进的迭代尺度法、梯度下降法、牛顿或拟牛顿法。**

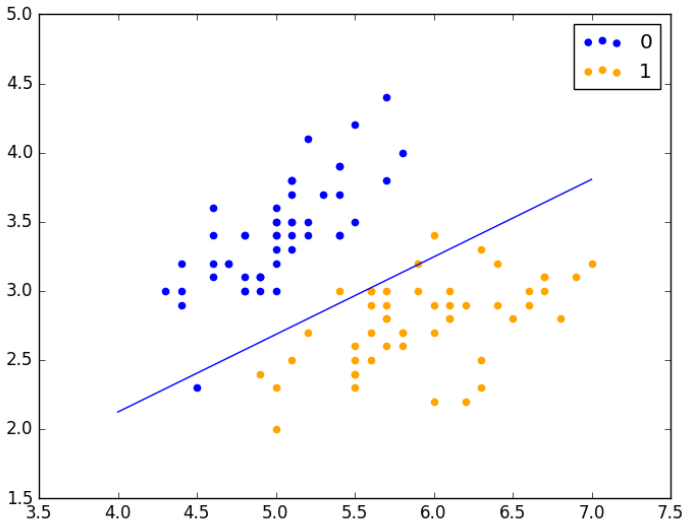
代码运行

在 iris 数据集上运行 LR.py 结果，可以看到：

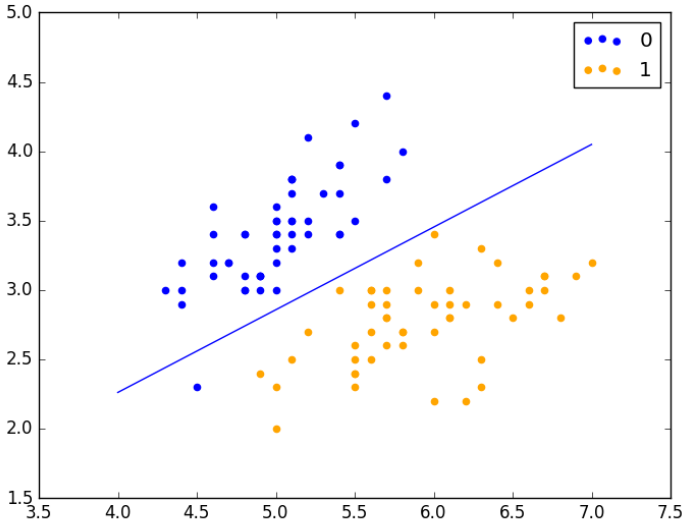
学习率为 0.01，迭代 10 次：



学习率为 0.01，迭代 30 次：



学习率为 0.01，迭代 60 次：


5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
0
1