Logistic Regression algorithm

詹雪莹，王耀威

November 7, 2016

1 软分类与逻辑回归

在分类中，为数据分配唯一的类别，这样称之为硬分类（Crisp Classification），而对应的称之为软分类（Soft Classification），每个数据都可能被分配多个可能的类别。

硬分类使用的是非概率模型，分类结果就是决策函数的结果，如PLA，决策函数为sign；软分类使用的是概率模型，输出不同类对应的概率，最后的分类结果取概率最大的类，如NB。

这次的逻辑回归是软分类算法，不是回归算法，之所以叫回归是因为这种分析方法来自于统计学中的回归分析。

逻辑回归，是一种常用的线性回归分析模型，如今已经广泛用在疾病自动诊断，经济预测等领域。数据既可以是连续的，也可以是离散的。然后通过逻辑回归分析，可以得到数据的权重，从而可以大致了解到底哪些因素是关键因素。同时根据该权值可以根据该因素预测目标的可能性。

2 逻辑回归算法

逻辑回归算法适用于二元分类，其类标签可以是+1,-1，也可以是1,2等，为了与理论课一致，这里采用0,1。不同的类标签对逻辑回归算法是有影响的，最直接的是梯度的更新公式略有不同，此外0,1的二元标签比较符合统计学的一些概念，0是negative类别，1是positive 的类别。所以本次实验会围绕类标签为0,1 进行推导说明，会涉及到一些基本的数学知识，这部分会用稍微严谨一些的数学理论说明，便于理解。但是还是会说明最基本的推导思想。

2.1 符号

符号说明如下：

Table 2.1: 符号

符号	描述
1	类别1（比如患有疾病）
0	类别2（比如不患有疾病）
$\boldsymbol{x}$	特征向量
y	预测结果
f	完美的目标函数
P	概率
N	已知的样本数目
$\boldsymbol{w}$	权重向量
s	特征的加权分数
θ	$logistic$函数
h	假说模型
g	最大熵模型（假说中似然性最大的h）
Err	模型误差
η	梯度下降的步长

注意：特征向量$\boldsymbol{x}$依然还是要补上第0维的常数1，以下不再说明了，原理与PLA一致!!!

2.2 模型建立与求解

一个软性的分类问题，目标函数定义如下：

$$f(\boldsymbol{x}) = P(1|\boldsymbol{x}) \in [0,1] \tag{2.1}$$

即，在给定特征向量$\boldsymbol{x}$的情况下，类比1出现的概率为多大。

其中特征$\boldsymbol{x}$根据问题不同而不同，为一个n 维欧几里得空间的特征向量，每一样例都可以欧几里得空间的一个向量来表示;1表示类别1，则用0 去表示类别2。

我们希望得到的理想数据如下：

$$(\boldsymbol{x_1}, y_1 = 0.6 = P(1|\boldsymbol{x_1}))$$

$$(\boldsymbol{x_2}, y_2 = 0.9 = P(1|\boldsymbol{x_2}))$$

.

.

.

$$(\boldsymbol{x_N}, y_N = 0.2 = P(1|\boldsymbol{x_N}))$$

但是现实中，我们是无法得到这种数据的，因为我们不可能知道某个类别（比如患病）的概率，只能知道最终是不是这个类别（是否患病），即

$$(\boldsymbol{x_1}, y_1 = 1 \sim P(1|\boldsymbol{x_1}))$$

$$(\boldsymbol{x_2}, y_2 = 1 \sim P(1|\boldsymbol{x_2}))$$

.

.

.

$$(\boldsymbol{x_N}, y_N = 0 \sim P(1|\boldsymbol{x_N}))$$

但是，它们服从$P(P(1|\boldsymbol{x_i})$的概率分布，我们可以用一些统计中最大熵的思想进行求解。

上面提到的最大熵原理是一种概率模型学习的一种准则。在最大熵理论下，概率模型的选取，在所有可能的概率模型（分布）中，熵值最大的模型为最好的模型，通常用一些约束条件来确定概率模型可能的取值集合，所有，最大熵原理也可以具体表达为满足约束条件的模型集合选取熵最大的模型的一种方法。

假设特征$\boldsymbol{x}$是一个d维的向量，即$\boldsymbol{x} = x_1, x_2, \ldots, x_d$，这些特征会对最终结果有直接的影响，我们模拟为一个带有权重的分数：

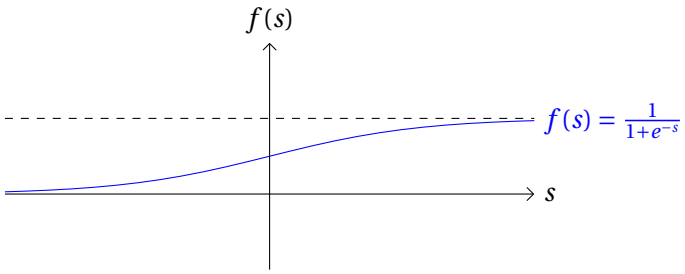
$$s = \sum_{i=0}^{d} = w_i x_i \tag{2.2}$$

其中，w_i为第i维特征的权重，$w_i > 0$表示该特征对类别1 (positive) 有正面影响，并且值越大，表示该特征对类别1 (positive)的贡献越大；相反的，$w_i < 0$ 表示该特征对类别1 (positive) 有负面影响，并且值越小，表示该特征对类别1的贡献越小；特别的，当$w_i = 0$的时候，表示该特征与类别1 (positive) 无关。s为最后的特征的加权分数。它的值越大，属于类别1 (positive) 的可能性越大；反之亦然。

由于我们需要得知类别1 (positive)的概率，所以我们将特征的加权分数s转换到估计的概率空间。为了合理的映射，我们采用逻辑函数（统计学叫法）或者生物生长曲线（生物学叫法）$logistic$ ($sigmoid$) 函数。如下：

$$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}} \tag{2.3}$$

它将$(-\infty, +\infty)$的空间压缩至$[0,1]$的概率空间，函数的图像如下：



这是一条平滑、单调的增函数，特别注意的是：

(1) $\theta(-\infty) = 0$指特征的加权分数越小，类别1 (positive)的概率为0

(2) $\theta(0) = 0.5$指特征的加权分数为中间值0，很难评估，即类别1 (positive)的概率为0.5

(3) $\theta(+\infty) = 1$指特征的加权分数越大，类别1 (positive)的概率为1

故基于$logistic$函数，我们可以用如下新的目标函数来近似之前的目标函数$f(\boldsymbol{x}) = P(y|\boldsymbol{x})$

$$h(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}} \tag{2.4}$$

$h(\boldsymbol{x})$为我们新的假说模型，要求解的参数为权重$\boldsymbol{w}$。

完美的目标函数是通过类别1 (positive)的概率来判断是否为类别1 (positive)，而假说模型只能通过否为类别1 (positive) 去求解概率值。根据统计的熵原理：在样本足够大的情况下，如果假说模型$h \approx$ 目标函数f，那么h 的似然性$\approx f$ 的概率值。即$likelihood(h) \approx probability(f)$，那么最优的假说模型为熵（似然性）最大的假说模型。

$$g = \underset{h}{\text{argmax}} \quad likelihood(h) \tag{2.5}$$

我们之前建立的假说，$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$表示类别1 (positive)的概率，那么类别2 (negative)的概率为$1 - h(\boldsymbol{x})$，故标签y为离散型随机变量，服从伯努利分布（二项分布）。

$$P(y|\boldsymbol{x}, \boldsymbol{w}) = h(\boldsymbol{x})^y (1 - h(\boldsymbol{x}))^{1-y} \tag{2.6}$$

根据贝叶斯法则，可得

$$\begin{aligned} likelihood(logistic \quad h) &= L(\boldsymbol{w}) \\ &\propto \prod_{n=1}^{N} P(y_n|\boldsymbol{x}_n, \boldsymbol{w}) \\ &= \prod_{i=1}^{N} h(\boldsymbol{x}_n)^{y_n}(1-h(\boldsymbol{x}_n))^{1-y_n} \end{aligned} \tag{2.7}$$

故我们只需要让此似然函数取得最大值即可

$$\max_{\boldsymbol{w}} \quad L(\boldsymbol{w}) \tag{2.8}$$

为了后续求解方便，我们将上述问题做一些变换，上面最大化的问题等价于如下的最小化的问题。

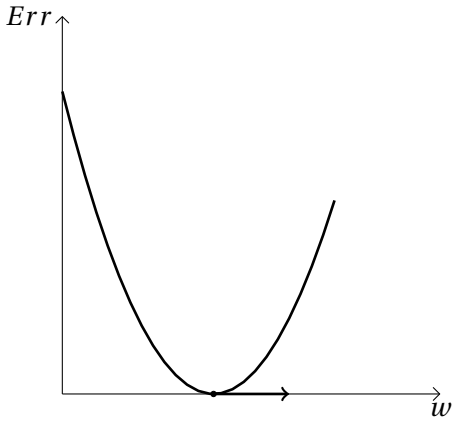
$$\min_{\boldsymbol{w}} \quad -\log L(\boldsymbol{w}) \tag{2.9}$$

主要做了两个处理，取对数不改变函数的极值点和最优解，添加了一个负号，将最大化问题转换为了一个最小化问题。这个新的误差函数在统计学叫做为交叉熵，交叉熵误差为

$$\begin{aligned} \min_{\boldsymbol{w}} \quad Err(\boldsymbol{w}) &= -\log \prod_{i=1}^{N} h(\boldsymbol{x}_n)^{y_n}(1-h(\boldsymbol{x}_n))^{1-y_n} \\ &= -\sum_{n=1}^{N} y_n \log(h(\boldsymbol{x}_n)) + (1-y_n)\log(1-h(\boldsymbol{x}_n)) \end{aligned} \tag{2.10}$$

log是以e为底。

由于误差函数$Err(\boldsymbol{w})$是一个连续可导，并且二阶可微的凸函数。根据凸优化理论，存在全局最优解，即为$\nabla Err(\boldsymbol{w}) = 0$



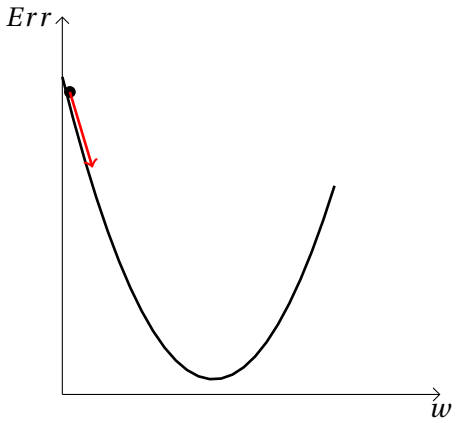
故我们首先要推导得到$\nabla Err(w_i)$，令$u=1+e^{-\boldsymbol{w}^T\boldsymbol{x}_n}$和$v=-\boldsymbol{w}^T\boldsymbol{x}_n$，则

$$\begin{aligned}\frac{\partial Err(w_i)}{\partial w_i} &= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{\partial \log(h(\boldsymbol{x}_n))}{\partial h(\boldsymbol{x}_n)}\right)\left(\frac{\partial h(\boldsymbol{x}_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right)+(1-y_n)\left(\frac{\partial \log(1-h(\boldsymbol{x}_n))}{\partial h(\boldsymbol{x}_n)}\right)\left(\frac{\partial h(\boldsymbol{x}_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right)\right] \\ &= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(\boldsymbol{x}_n)}\right)+(1-y_n)\left(\frac{-1}{1-h(\boldsymbol{x}_n)}\right)\right]\left[\left(\frac{-1}{u^2}\right)(e^v)(-x_{n,i})\right] \\ &= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(\boldsymbol{x}_n)}\right)-(1-y_n)\left(\frac{1}{1-h(\boldsymbol{x}_n)}\right)\right][h(\boldsymbol{x}_n)(1-h(\boldsymbol{x}_n))](x_{n,i}) \\ &= -\sum_{n=1}^{N}\left[(y_n)(1-h(\boldsymbol{x}_n))-(1-y_n)h(\boldsymbol{x}_n))\right](x_{n,i}) \\ &= -\sum_{n=1}^{N}(y_n-h(\boldsymbol{x}_n))(x_{n,i})\end{aligned} \tag{2.11}$$

故$Err(w_i)$的梯度如下：

$$\nabla Err(w_i)=\sum_{n=1}^{N}\left(\frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}}-y_n\right)(x_{n,i}) \tag{2.12}$$

不幸的是，这个梯度表达式为一个非线性函数，故要求解函数零点非常困难，故我们采用了迭代最优化的方式去求解。由于$Err(\boldsymbol{w})$是一个凸函数，故我们只要沿着梯度下降的方向去更新求解$\boldsymbol{w}$，就一定能较迅速找到最优解，因为梯度是函数变化最快的方向。



假设第t步我们已经得到了权重$\boldsymbol{w}_t$，那么，根据梯度下降法，我们可以得到如下更新公式：

$$\boldsymbol{w}_{t+1}=\boldsymbol{w}_t-\eta\nabla Err(\boldsymbol{w}_t) \tag{2.13}$$

其中，$\eta>0$表示梯度下降的步长，为人工设置参数。

于是逻辑回归算法去求解分类问题如下：

Algorithm 1 逻辑回归训练算法

输入: 特征向量集合$\{\boldsymbol{x}\}$和标签集合$\{y\}$

输出: 最优解$\boldsymbol{w}_{t+1}$

初始化: 随机初始化$\boldsymbol{w}_0$

$for \quad t = 0, 1, ...$

1. 通过公式2.12 计算每一个维度的梯度

   $for \quad i = 0, 1, ..., d$

$$\nabla Err(w_{t,i}) = \sum_{n=1}^{N} \left(\frac{1}{1 + e^{-\boldsymbol{w}_t^T \boldsymbol{x}_n}} - y_n \right) (x_{n,i})$$

2. 通过公式2.13迭代更新权重的每一个维度

   $for \quad i = 0, 1, ..., d$

$$w_{t+1,i} = w_{t,i} - \eta \nabla Err(w_{t,i})$$

直到$\nabla Err(\boldsymbol{w}) = 0$或者迭代足够多次

由于我们模型的目标函数为一个光滑的凸函数，故我们有很多可以优化数值计算的方法，保证全局最优解，常用的改进方式可以为牛顿法，拟牛顿法，共轭梯度等。除此之外由于每一次更新都需要重新计算整个训练集的梯度，如果是大数据这种方式就很慢了，相应的办法为随机梯度下降（SGD），上述提到的改进就不在这里详细展开说明了。

故我们最终可以得到特征权重w_{t+1}，它是基于已有数据集产生的加权参数。接下来只需要用这个参数去进行预测分类。对于一个测试样本$\boldsymbol{x}_{test}$，计算它属于类别1（positive）的概率$\frac{1}{1+e^{-\boldsymbol{w}^T \boldsymbol{x}_{test}}}$。如果该值大于0.5即为类别1（positive），否则就是类别2（positive）。

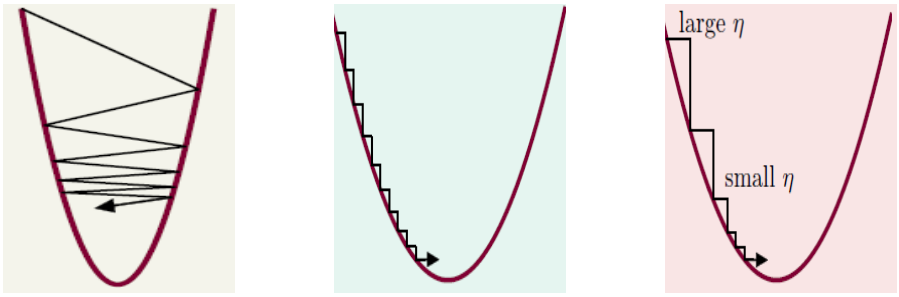
2.3 学习速率（梯度步长）

如果不采用共轭梯度法去进行梯度迭代，必然就必须考虑学习速率如何设置的问题。梯度下降步长η会直接决定迭代求解能否求解到梯度的全局最优值，设置过大或者过小都不是很好。

如果过大，模型求解过程比较震荡，很有可能无法求得全局最优解；

如果过小，模型一定可以求得最优解，但是效率比较低；

一般没有通用的办法和理论去确定，只有一些经验公式和手动调节的方式，我给出一种较为接受的方法。我们可以结合两者达到最优效果。在刚开始的时候设置一个较大



(a) η过大　　(b) η过小　　(c) η动态调节

Figure 2.1: 学习速率η的影响

的步长，随着迭代次数的增多，在较接近极值点（变化率很小）时，将步长调节为一个较小值，这种动态调节方式为最优方案，如上图。

除此之外还可以通过验证集的方式确定学习速率，这个之前有讲过，这里在重复一下：

如果在给定数据充足的情况下，可以将已有数据随机分成2分，一份用于模型的拟合（训练集），一份用于模型的选择（验证集），设置不同的学习速率都可以较好拟合数据的时候，选择验证集效果最好的一个为最优模型。如下图：

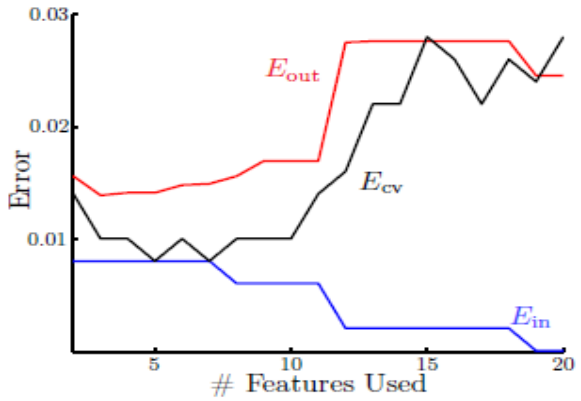


Figure 2.2: 数据影响

其中E_{in}表示训练时候的误差，E_{out}表示预测时候的误差，E_{cv} 表示验证时候的误差。可以通过上图分析发现，验证误差（E_{cv}）比较接近预测误差（E_{out}）。通过验证的方式，我们可以提高我们模型的稳定性，和对未知数据的预测能力。具体原因牵扯到数据抽样，霍夫丁不等式和VC维度等知识，这里不深究了，有致力于做机器学习研究的同学可以下去好好研究一下。

3 一个简单的例子

数据集如下：

编号	特征1	特征2	标签
train1	1	-1	1
train2	3	3	0
test1	-2	3	?

初始化$\mathbf{w}_0 = \{1,-2,3\}$，设置学习速率为1且只迭代1次。
可以先得到权重分数：

$$s_1 = 1*1+(-2)*1+3*(-1) = -4$$

$$s_2 = 1*1+(-2)*3+3*3 = 4$$

用公式计算每一维的梯度：

$$\nabla Err(w_{t,i}) = \sum_{n=1}^{N}\left(\frac{1}{1+e^{-\boldsymbol{w}_t^T\boldsymbol{x}_n}} - y_n\right)(x_{n,i})$$

第0维：

$$\nabla Err(w_{0,0}) = \left(\frac{1}{1+e^{-(-4)}} - 1\right)(1) + \left(\frac{1}{1+e^{-4}} - 0\right)(1) = 0$$

第1维：

$$\nabla Err(w_{0,1}) = \left(\frac{1}{1+e^{-(-4)}} - 1\right)(1) + \left(\frac{1}{1+e^{-4}} - 0\right)(3) = 1.964$$

第2维：

$$\nabla Err(w_{0,2}) = \left(\frac{1}{1+e^{-(-4)}} - 1\right)(-1) + \left(\frac{1}{1+e^{-4}} - 0\right)(3) = 3.928$$

用公式更新每一个维度的权重

$$w_{t+1,i} = w_{t,i} - \eta\nabla Err(w_{t,i})$$

第0维：

$$w_{1,0} = w_{0,0} - \eta\nabla Err(w_{0,0}) = 1 - 1*0 = 1$$

第1维：

$$w_{1,1} = w_{0,1} - \eta\nabla Err(w_{0,1}) = -2 - 1*1.964 = -3.964$$

第2维：

$$w_{1,2} = w_{0,2} - \eta\nabla Err(w_{0,2}) = 3 - 1*3.928 = -0.928$$

故更新后$\mathbf{w}_1 = \{1, -3.964, -0.928\}$，停止学习，进行预测

$P(1|test1, \mathbf{w}_1) = \frac{1}{1+e^{-(1*1+(-3.964)*(-2)+(-0.928)*3)}} = 0.9979$

故test1的预测标签为1

4 数据集与评测指标

本次实验的结果评测，采用4种指标，综合评测，分别为**Accuracy（准确率），Precision（正确率），Recall（召回率），F1（F1值）**，之前详细讲过，这里不在赘述。

数据集采用‘Breast Cancer Wisconsin (Diagnostic) Data Set”,乳腺癌诊断数据集。9维特征向量，标签为0（不患病）1（患病）。一共包括两个文件：

train.csv: 583个样例，每一个样例为9维特征向量+一个标签

test.csv: 100个样例，每一个样例为9维特征向量+一个标签

5 实验任务与提交要求

本次实验只需要实验二元分类即可，需要完成的任务如下：

(1) 在提供的数据集上实现逻辑回归，具体要求请看实验要求细则

(2) 采用4种评测指标评价你的实验结果

(3) 尽可能的优化与分析

提交要求与原来一致

报告：

- 分类结果分析与展示
- 4种指标的数据分析
- 实验思路（推荐伪代码）
- 详细描述创新点（如果有的话）及优化前后对比
- 命名格式：学号_拼音名字.pdf，后缀取决于编程语言

代码：

- 同一种算法只需提交一份，例如你有多个逻辑回归版本提交最优的即可

- 命名格式：学号_拼音名字.xxx，后缀取决于编程语言

注意事项：

- 截止时间：2016年11月20号23:59:59前提交FTP对应文件夹，否则视为迟交
- FTP 地址: ftp://my.ss.sysu.edu.cn/~ryh
- 任何形式的抄袭，双方成绩均为0