实验五：Logistic Regression algorithm
——逻辑回归

PPT制作：黄行昌，林东定
出题人　：詹雪莹，王耀威

软分类和逻辑回归

- 根据位数据是否分配唯一的类别可分为：
  - 硬分类：非概率模型，由决策函数决定
    - 如：决策树，PLA
  - 软分类：概率模型，最终取概率最大的类
    - 如：NB

- 逻辑回归是软分类算法，广泛运用在疾病自动诊断，经济预测等领域。通过回归分析，可以得到数据权重，从而可以大致了解哪些因素是关键因素，也可以根据权重了解预测目标的可能性

逻辑回归算法

- 理论推导
  - 符号规定

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）
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
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

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 对于一个软性分类问题，目标函数定义如下

$$F(x) = P(1|x)\epsilon[0,1]$$

      - 即在给定特征向量x的情况下，类别1出现的概率为多大
    - 我们希望得到的理想数据如下

$$(x_1, y_1 = 0.6 = P(1|x_1))$$
$$(x_2, y_2 = 0.9 = P(1|x_2))$$
$$\ldots\ldots$$
$$\ldots\ldots$$
$$(x_N, y_N = 0.2 = P(1|x_N))$$

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 但是现实中我们可以得到的数据只能是唯一的

$$(x_1, y_1 = 1 \sim P(1|x_1))$$
$$(x_2, y_2 = 1 \sim P(1|x_2))$$
$$\ldots\ldots$$
$$\ldots\ldots$$
$$(x_N, y_N = 0 \sim P(1|x_N))$$

    - 但是，他们服从$\mathrm{P}\big(P(1|x_i)\big)$的概率分布，我们可以用一些统计学中最大熵的思想进行求解
      - 最大熵：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，所以，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。

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 假设特征向量$x = x_1, x_2, \dots\dots, x_d$，这些特征会对最终结果有直接的影响，我们模拟一个带权重的分数

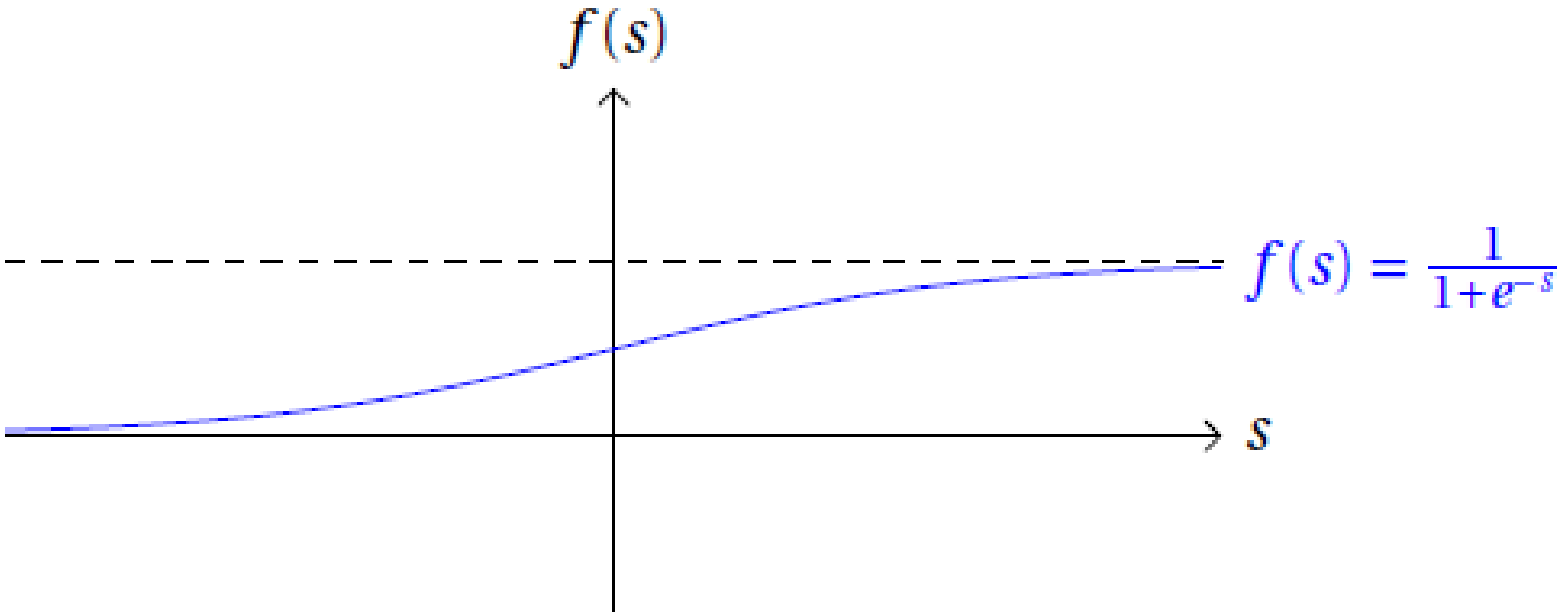
$$s = \sum_{t=0}^{d} = w_i x_i$$

      - 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，反之亦然
      - 最终s的值越大，属于类别1(positive)的可能性越大
    - 由于我们需要得知类别1(positive)的概率，所以将加权分数s转换到另一个合理的映射空间，叫logistic(sigmoid)函数

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

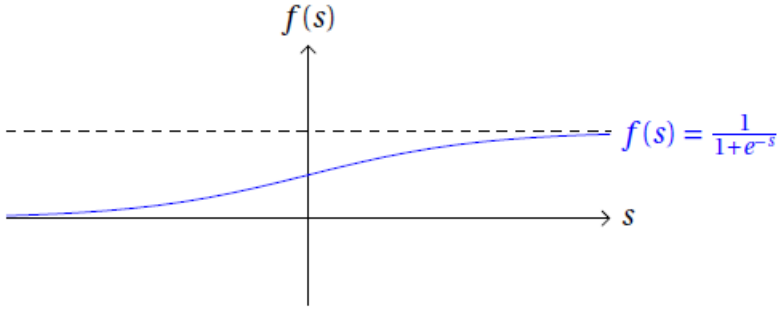
逻辑回归算法

- 理论推导
  - 模型建立与求解
    - Sigmoid函数把$(-\infty,+\infty)$空间压缩到[0,1]的概率空间



逻辑回归算法

- 理论推导
  - 模型建立与求解



- Sigmoid函数有几个特征

① $\theta(-\infty) = 0$，特征向量的加权分数无穷小，该特征向量为类别1(positive)的概率为0

② $\theta(0) = 0.5$，特征向量的加权分数为0，很难评估，类别1(positive)的概率为0.5

③ $\theta(+\infty) = 0$，特征向量的加权分数无穷大，该特征向量为类别1(positive)的概率为1

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 再结合logistic函数，我们可以用下面新的目标函数来近似之前的目标函数$f(x) = P(y|x)$

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

      - $h(x)$为我们新的假说模型，要求解的为权重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(postive)，而假说模型只能根据是否为类别1(positive)来求解概率值。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$likelihood(h) \approx probability(f)$

逻辑回归算法

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

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

    - 根据贝叶斯法则，可得

$$likelihood(logistic\ h) = L(w)$$
$$\propto \prod_{n=1}^{N} P(y_n|x_n, w)$$
$$= \prod_{t=1}^{N} h(x_n)^{y_n} (1 - h(x_n))^{1-y_n}$$

逻辑回归算法

- 理论推导
  - 模型建立与求解

$$likelihood(logistic\ \ h) = L(w)$$
$$\propto \prod_{n=1}^{N} P(y_n|x_n, w)$$
$$= \prod_{t=1}^{N} h(x_n)^{y_n}(1-h(x_n))^{1-y_n}$$

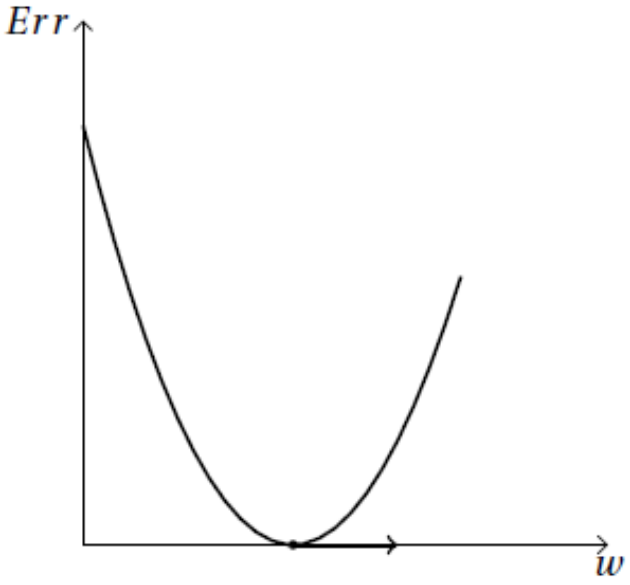
  - 故我们只需要让此似然函数取得最大值即可 $\max_{w} L(w)$
  - 为了后续求解方便，我们将上述问题做一些变换，求最大化的问题等价于求最小化的问题 $\min_{w} -logL(w)$
    - 主要做了两个处理，取对数不改变函数的极致点和最优解，添加负号，将最大化问题转换成最小化问题

$$\min_{w} Err(w) = -log \prod_{i=1}^{N} h(x_n)^{y_n}(1-h(x_n))^{1-y_n}$$
$$= -\sum_{n=1}^{N} y_n \log\big(h(x_n)\big) + (1-y_n)\log(1-h(x_n))$$

log以e为底

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 由于误差函数$Err(w)$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$\nabla Err(w) = 0$



log以e为底

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 故我们首先要推导得到$\nabla Err(w_i)$，令$u = 1 + e^{-w^T x_n}$和$v = -w^T x_n$，则

$$\frac{\partial Err(w_i)}{\partial w_i} = -\sum_{n=1}^{N}\left[(y_n)\left(\frac{\partial \log(h(x_n))}{\partial h(x_n)}\right)\left(\frac{\partial h(x_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right) + (1-y_n)\left(\frac{\partial \log(1-h(x_n))}{\partial h(x_n)}\right)\left(\frac{\partial h(x_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right)\right]$$

$$= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(x_n)}\right) + (1-y_n)\left(\frac{-1}{1-h(x_n)}\right)\right]\left[\left(\frac{-1}{u^2}\right)(e^v)(-x_{n,i})\right]$$

$$= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(x_n)}\right) - (1-y_n)\left(\frac{1}{1-h(x_n)}\right)\right]\left[h(x_n)(1-h(x_n))\right](x_{n,i})$$

$$= -\sum_{n=1}^{N}\left[(y_n)(1-h(x_n)) - (1-y_n)h(x_n)\right](x_{n,i})$$

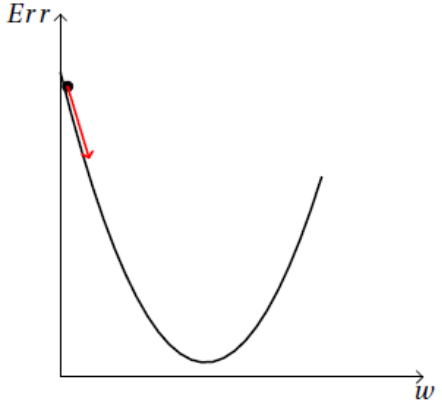
$$= -\sum_{n=1}^{N}(y_n - h(x_n))(x_{n,i})$$

逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 故$Err(w_i)$的梯度为

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

      - 然而不幸的是，这个表达式是一个非线性函数，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法去求解。由于$Err(w)$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w，就一定能较迅速的找到最优解（梯度是函数变化最快的方向）



逻辑回归算法

- 理论推导
  - 模型建立与求解
    - 假设第t步我们已经得到了权重w_t，那么根据梯度下降法，我们可以得到如下更新公式

$$w_{t+1} = w_t - \eta \nabla Err(w)$$

      - 其中，η>0表示梯度下降的步长，为人工设置的参数
    - 于是逻辑回归算法求解分类问题算法如下
      - 输入：特征向量集合{x}和标签集合{y}
      - 输出：最优解w_{t+1}
      - 初始化：随机初始化w_0
      - 通过梯度公式计算每一个维度的梯度

$$\nabla Err(w_{t,i}) = \sum_{n=1}^{N} \left(\frac{1}{1+e^{-w_t^T x_n}} - y_n \right) (x_{n,i}) \ \ (for\ t = 0,1, \dots \dots, d)$$

      - 通过公式迭代更新权重的每一个维度

$$w_{t+1,i} = w_{t,i} - \eta \nabla Err(w_{t,i}) \ \ (for\ t = 0,1, \dots \dots, d)$$

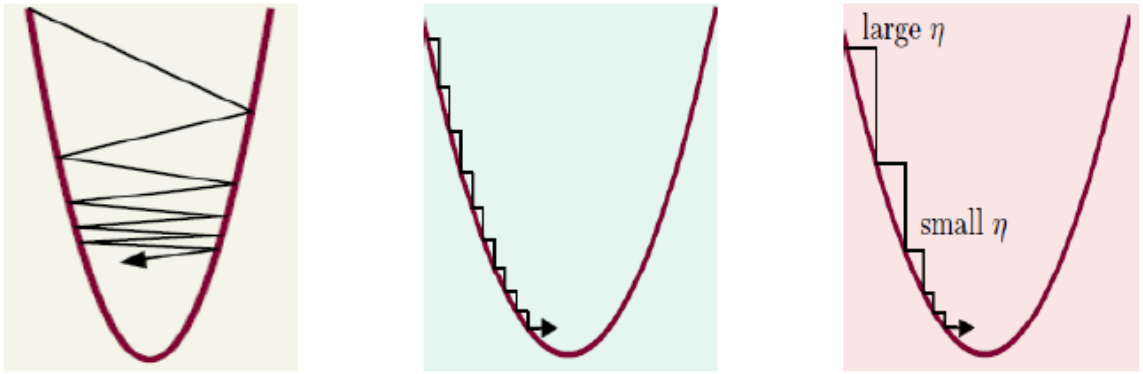
      直到$\nabla Err(w) = 0$或者迭代足够多次
      - 通过基于已有数据集产生的加权参数w_{t+1}预测分类，计算属于类别1(positive)的概率

逻辑回归算法

- 理论推导
  - 学习速率（梯度步长）
    - 梯度步长η的设置会直接影响迭代解能否求解到梯度的全局最优值
    - η设置过大，求解过程比较震荡，很可能无法求出最优解
    - η设置过小，模型一定可以求得最优解，但是效率比较大

逻辑回归算法

- 理论推导
  - 学习速率（梯度步长）
    - 一般没有通用的办法和理论确定，这里给出一种方法结合两者达到最优效果：在刚开始时设置一个较大的步长，随着迭代次数增多，在较接近极值点（变化率很小时），将步长调整为一个较小值。



(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

逻辑回归算法

- 理论推导
  - 学习速率（梯度步长）
    - 除此之外还可以通过验证集的方式确定学习率
    - 在给定数据充足的情况下，可以将已有数据随机分成两份，一份用于模型拟合（训练），另一部分用于模型选择（验证集），设置不同的学习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。

逻辑回归算法

- 理论推导
  - 学习速率（梯度步长）
    - E_{in}：训练误差，
      E_{out}：预测误差
      E_{cv}：验证误差

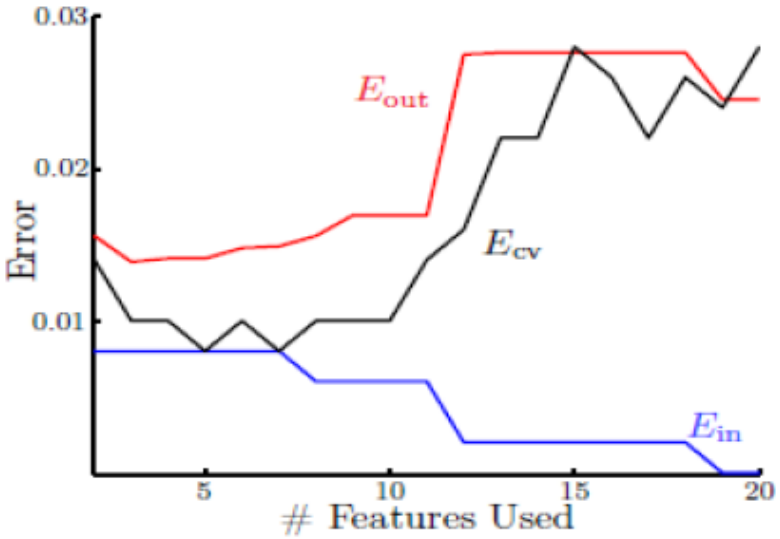


Figure 2.2: 数据影响

- 通过右图分析，
  发现验证误差比较接近预测误差，
  因此通过验证的方式，可以提高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，VC维等知识

逻辑回归算法

- 一个简单的例子
  - 数据集如下，初始化$w_0 = \{1, -2, 3\}$，设置学习速率为1且只迭代1次：
  - 计算权重分数
  - 计算每一维的梯度
  - 更新每一维的权重

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

逻辑回归算法

- 一个简单的例子

1. 计算每一个样例的权重分数

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

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

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

2. 计算每一维的梯度

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

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

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

逻辑回归算法

- 一个简单的例子

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

3. 计算每一维的权重

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

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

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

4. 迭代一次，停止学习，进行预测$w_1 = \{1, -3.964, -0.928\}$

$P(1|test1, w_1) = \frac{1}{1+e^{-(1*1+(-3.964)*(-2)+(-0.928)*3)}} = 0.9979 > 0.5$

因此test1预测标签为1

数据集与评测指标

本次数据集为乳腺癌诊断，标签为0（不患病）1（患病）

train.csv: 583个样例，每个样例为9维的特征向量+一个标签

test.csv: 100个样例，每个样例为9维的特征向量+一个标签

实验任务与提交要求

任务：

1. 在提供的数据集上实现逻辑回归，具体请看“lab5实验要求”
2. 采用4种评测指标评价你的实验结果
3. 尽可能地优化与分析

实验任务与提交要求

提交要求与原来一致：

报告篇

1. 结果分析与展示
2. 4种评测指标的数据分析
3. 实验思路（推荐伪代码）
4. 详细描述创新点（如果有）及优化前后对比
5. 命名：学号_拼音名字.pdf

实验任务与提交要求

提交要求：

代码篇

1. 只需要提交一份逻辑回归的代码，假如有多个版本的逻辑回归，请选择你认为最优的提交。
2. 命名：学号_拼音名字.xxx 提交，后缀取决于编程语言

实验任务与提交要求

提交要求：

注意事项

1. 截止日期：2016年11月20日23点59分59秒，超过则视为迟交
2. FTP地址：ftp://my.ss.sysu.edu.cn/~ryh
3. 抄袭，双方均0分