北京交通大学硕士研究生《机器学习》课件

第12章 对数线性分类模型

北京交通大学《机器学习》课程组

分类算法的影响因素

- 类内部表示的各个部分对分类算法都有重要影响
  - 类的认知表示
    - ✓ 神经网络：不可见的复杂回归函数
    - ✓ K近邻：类的外延表示
    - ✓ 线性分类模型：线性函数
  - 类相似性映射
- 求解路径
- 最优化算法

对数线性分类模型的引入

- 在已知(X,U)、X=Y 且c > 1的情况下，如果预知c个类的类认知表示形式，有时候人们希望其类相似性函数位于[0,1] 之间，以与人们的直观保持一致。由于简单的线性分类模型没有这样约束类相似性映射，因此不满足需求，需要重新设计分类算法。
- 满足这个要求的两个线性分类模型
  - ✓ Softmax回归
  - ✓ Logistic回归

目录

- 12.1 Softmax回归
- 12.2 Logistic回归
- 总结

激活函数

- Heaviside (or step/threshold) function: output binary

$$\phi(z_i) = \begin{cases} 0, & z_i < 0 \\ 1, & z_i \geq 0 \end{cases}$$

- Sigmoid function: output continuous, $0 \leq y_i(\boldsymbol{x}) \leq 1$.

$$\phi(z_i) = \frac{1}{1 + \exp(-z_i)}$$

- Softmax function: output $0 \leq y_i(\boldsymbol{x}) \leq 1$, $\sum_{i=1}^{n} y_i(\boldsymbol{x}) = 1$.

$$\phi(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{n} \exp(z_j)}$$

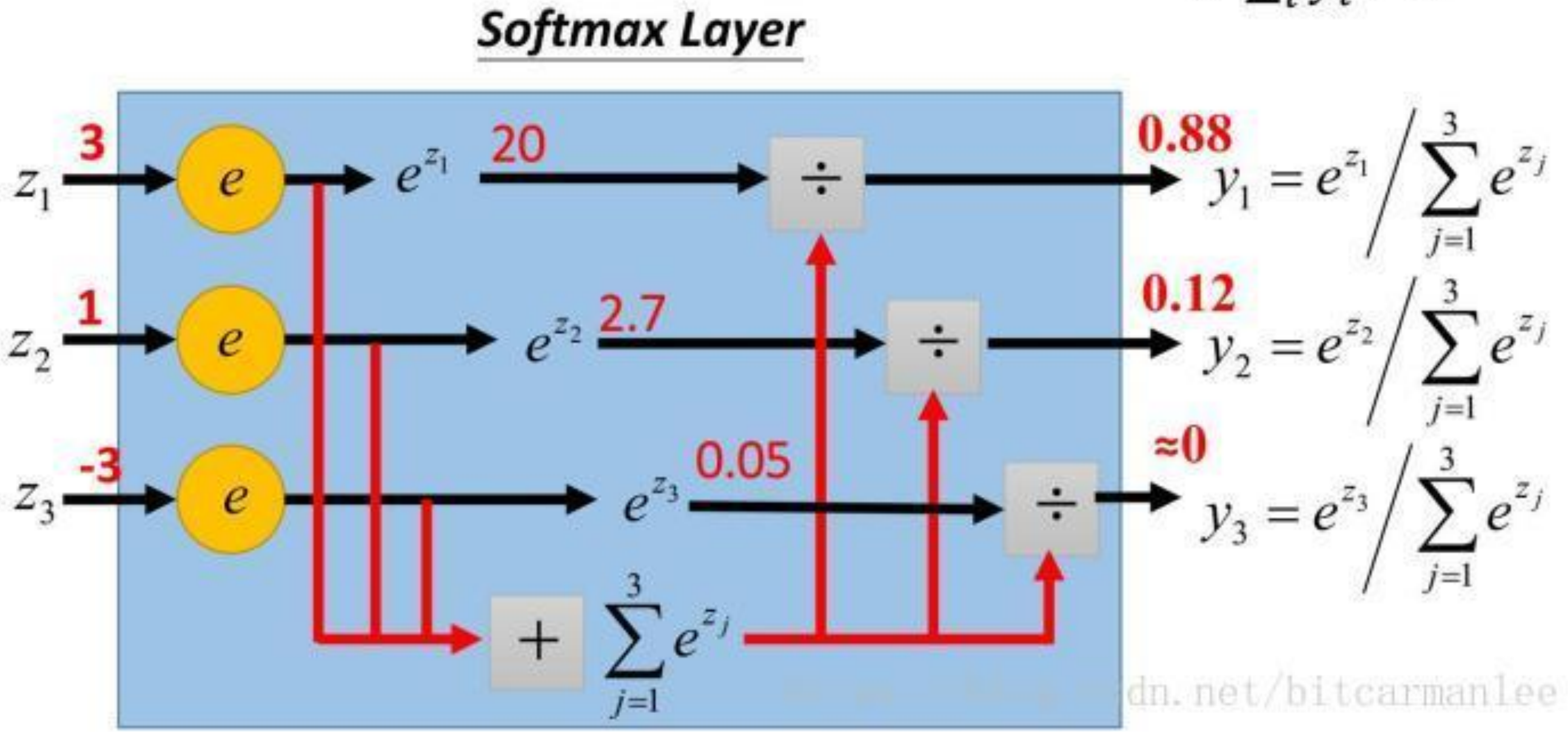
- Hyperbolic tan function: output continuous, $-1 \leq y_i(\boldsymbol{x}) \leq 1$.

$$\phi(z_i) = \frac{\exp(z_i) - \exp(-z_i)}{\exp(z_i) + \exp(-z_i)}$$

Softmax作为输出层

- Softmax layer as the output layer

Probability:
- $1 > y_i > 0$
- $\sum_i y_i = 1$



简单的线性分类模型

类认知表示为：$\forall i, \underline{Y_i} = (x, w_i^T x + w_{i0})$

类相似性映射为：$\forall k, \mathrm{Sim}_Y(x_k, \underline{Y_i}) = \exp(w_i^T x_k + w_{i0})$

✖ $\forall k \forall i, \mathrm{Sim}_Y\left(x_k, \underline{Y_i}\right) \in [0, 1]$

归一化

$$\forall k \forall i, \mathrm{Sim}_Y\left(x_k, \underline{Y_i}\right) = p\left(\underline{Y_i} \middle| x_k\right) = \frac{\exp(w_{i0} + w_i^T x_k)}{\sum_{j=1}^{c} \exp(w_{j0} + w_j^T x_k)} \in [0, 1]$$

目标函数

- 依据紧致性原则，最大化目标函数为：

$$L=\prod_{k=1}^{N} Sim_Y(x_k,\underline{Y_{\vec{x}_k}})=\prod_{k=1}^{N}\prod_{i=1}^{c} Sim_Y(x_k,\underline{Y_i})^{u_{ik}}$$

$$U=[u_{ik}]_{c\times N},\forall i\forall k,u_{ik}\in\{0,1\},\ \sum_{i=1}^{c}u_{ik}=1$$

- 为简化计算，将连乘转化为求和：对目标函数取对数

$$\begin{aligned}\ln L&=\sum_{k=1}^{N}\ln Sim_Y(x_k,\underline{Y_{\vec{x}_k}})\\&=\sum_{k=1}^{N}\sum_{i=1}^{c}u_{ik}\ln Sim_Y(x_k,\underline{Y_i})\\&=\sum_{k=1}^{N}\sum_{i=1}^{c}u_{ik}(w_{i0}+w_i^T x_k-\ln\sum_{j=1}^{c}\exp(w_{j0}+w_j^T x_k))\end{aligned}$$

利用梯度下降法求解

通过梯度下降法进行求解，关于w_i的偏导数为

$$\frac{\partial \ln L}{\partial w_i} = \sum_{k=1}^{N} \left(u_{ik} x_k - \frac{\exp(w_i^T x_k + w_{i0})}{\sum_{i=1}^{c} \exp(w_i^T x_k + w_{i0})} x_k \right)$$

$$= \sum_{k=1}^{N} x_k \left(u_{ik} - Sim_Y(x_k, \underline{Y_i}) \right)$$

关于w_{i0}的偏导数为

$$\frac{\partial \ln L}{\partial w_{i0}} = \sum_{k=1}^{N} \left(u_{ik} - \frac{\exp(w_i^T x_k + w_{i0})}{\sum_{i=1}^{c} \exp(w_i^T x_k + w_{i0})} \right)$$

$$= \sum_{k=1}^{N} \left(u_{ik} - Sim_Y(x_k, \underline{Y_i}) \right)$$

不可能有闭式解，需要学习率，通常不采用这种方法

Newton-Raphson 算法求解

为了求出$\forall i, w_{i0}, w_i$可以采用Newton-Raphson算法，这要求计算

$$\frac{\partial^2 \ln L}{\partial w_i \partial w_j} = -\sum_{k=1}^{N} x_k x_k^T Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

$$\frac{\partial^2 \ln L}{\partial w_i \partial w_{j0}} = -\sum_{k=1}^{N} x_k Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

$$\frac{\partial^2 \ln L}{\partial w_{i0} \partial w_{j0}} = -\sum_{k=1}^{N} Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

计算Hessian矩阵

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_{11}} & \boldsymbol{H_{11}} & \cdots & \boldsymbol{H_{1c}} \\ \boldsymbol{H_{21}} & \boldsymbol{H_{22}} & \cdots & \boldsymbol{H_{2c}} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{H_{c1}} & \boldsymbol{H_{c2}} & \cdots & \boldsymbol{H_{cc}} \end{pmatrix} \quad \boldsymbol{H_{ij}} = \begin{pmatrix} \dfrac{\partial^2 lnL}{\partial w_{i0} \partial w_{j0}} & \left(\dfrac{\partial^2 lnL}{\partial w_{i0} \partial w_j}\right)^T \\ \dfrac{\partial^2 lnL}{\partial w_i \partial w_{j0}} & \dfrac{\partial^2 lnL}{\partial w_i \partial w_j} \end{pmatrix}$$

Newton-Raphson 算法求解

令$\beta = (\beta_1, \ldots, \beta_c)^T$

$\beta_i = (w_{i0}, (w_i)_1, \ldots, (w_i)_p)^T$

$(w_i)_p$ 表示向量w_i的第p个分量

牛顿-拉夫森迭代法需求出二阶导，Hessian矩阵存不存在，跟数据集有关。有一个好处，两步更新，没有学习率。Hessian矩阵可逆，就没有学习率。二阶导为0，就用一阶导，梯度下降法。

由此得到在Softmax 回归算法中，β的更新迭代公式为

$$\beta \leftarrow \beta - \mathrm{H}^{-1}\frac{\partial L}{\partial \beta}$$

$$\frac{\partial L}{\partial \beta} = \left(\frac{\partial L}{\partial \beta_1}, \frac{\partial L}{\partial \beta_2}, \cdots, \frac{\partial L}{\partial \beta_c}\right)^T \qquad \frac{\partial L}{\partial \beta_i} = \left(\frac{\partial L}{\partial w_{i0}}, \left(\frac{\partial L}{\partial w_i}\right)^T\right)$$

Softmax冗余问题

C组参数$\beta_1, \beta_2, \cdots, \beta_c$中有一组是冗余的，满足$\max_\beta \ln L$的解不唯一。

$$Sim_Y(x_k, \underline{Y_i}) = \frac{\exp(w_{i0} + w_i^T x_k)}{\sum_{j=1}^{c} \exp(w_{j0} + w_j^T x_k)}$$

分子分母同时乘以

$\exp(-w_{c0} - w_c^T x_k)$

$$= \frac{\exp(w_{i0} + w_i^T x_k - w_{c0} - w_c^T x_k)}{\sum_{j=1}^{c} \exp(w_{j0} + w_j^T x_k - w_{c0} - w_c^T x_k)}$$

$$= \frac{\exp(w_{i0} - w_{c0} + (w_i - w_c)^T x_k)}{1 + \sum_{j=1}^{c-1} \exp(w_{j0} - w_{c0} + (w_j - w_c)^T x_k)}$$

参数独立的有c-1组，说明softmax模型被过度参数化了，这样解存在无穷多组解。

Softmax回归的新目标函数

奥卡姆剃刀　　在这些解中找到一个最简单的

定义β的复杂度$D(\beta)$，复杂度也要达到最小。$D(\beta)=\|\beta\|^2=\sum_{i=1}^{c}\sum_{j=0}^{p}w_{ij}^2$

最大化softmax回归新的目标函数

$$\begin{aligned}
\ln L-\frac{\lambda}{2}\mathrm{D}(\beta) &= \sum_{k=1}^{N}\ln Sim_Y(x_k,\underline{Y_{\vec{x}_k}})-\frac{\lambda}{2}D(\beta)\\
&=\sum_{k=1}^{N}\sum_{i=1}^{c}u_{ik}\ln Sim_Y(x_k,\underline{Y_i})-\frac{\lambda}{2}D(\beta)\\
&=\sum_{k=1}^{N}\sum_{i=1}^{c}u_{ik}(w_{i0}+w_i^Tx_k-\ln\sum_{j=1}^{c}\exp(w_{j0}+w_j^Tx_k))-\frac{\lambda}{2}\sum_{i=1}^{c}\sum_{j=0}^{p}w_{ij}^2
\end{aligned}$$

引入了超参数λ>0

优点：当特征维数 p 远远大于样本个数 N 时，12.2式的Hessian矩阵不可逆， β更新迭代公式就不能使用了。这里Hessian矩阵是永远可逆的，因此用**Newton-Raphson算法**永远是可行的。

12.2 Logistic回归

- **也是采用判别式的思想，输出类认知表示用函数表示，但是与判别函数法的区别在于：**
  - 有一类的类输出认知表示未显示表达
  - 每个输出类的类相似性映射是逻辑斯谛分布的密度函数

- **类认知表示：** $x_k \in X_i$，则$u_{ik}=1$，且$\forall j \neq i, u_{jk}=0$

  当$1 \le i \le c-1$　$\underline{Y_i}=(x, w_i^T x + w_{i0})$

  第c类的输出类认知表示未知

- **类相似性映射：**

当$1 \le i \le c-1$

$$Sim_Y(x,\underline{Y_i}) = p(\underline{Y_i} \mid x) = \frac{\exp(w_i^T x + w_{i0})}{1+\sum_{i=1}^{c-1}\exp(w_i^T x + w_{i0})}$$

$i=c$

$$Sim_Y(x,\underline{Y_c}) = p(\underline{Y_c} \mid x) = \frac{1}{1+\sum_{i=1}^{c-1}\exp(w_i^T x + w_{i0})}$$

目标函数

依据类紧致性准则最大化目标函数：

$$\max_{\underline{Y_1},\underline{Y_2},\ldots,\underline{Y_{c-1}}} L = \prod_{k=1}^{N} Sim_Y(x_k, \underline{Y_{\vec{x}_k}})$$

$$= \prod_{k=1}^{N}\prod_{i=1}^{c} Sim_Y(x_k, \underline{Y_i})^{u_{ik}}$$

$$= \prod_{k=1}^{N}\prod_{i=1}^{c} p(\underline{Y_i} \mid x)^{u_{ik}}$$

$x_k \in X_i$，则$u_{ik} = 1$，且$\forall j \neq i, u_{jk} = 0$

取对数

$$\max_{\underline{Y_1},\underline{Y_2},\ldots,\underline{Y_{c-1}}} \ln L = \prod_{k=1}^{N}\prod_{i=1}^{c} u_{ik} Sim_Y(x_k, \underline{Y_i})$$

$$= \prod_{k=1}^{N}\prod_{i=1}^{c} u_{ik}(w_i^T x_k + w_{i0}) - \sum_{k=1}^{N} \ln(1 + \sum_{i=1}^{c-1} \exp(w_i^T x_k + w_{i0}))$$

梯度下降法求解

关于w_i 的偏导数为

$$\frac{\partial \ln L}{\partial w_i}=\sum_{k=1}^{N}(u_{ik}x_k-\frac{\exp(w_i^T x_k+w_{i0})}{1+\sum_{i=1}^{c-1}\exp(w_i^T x_k+w_{i0})}x_k)$$

$$=\sum_{k=1}^{N}x_k(u_{ik}-Sim_Y(x_k,\underline{Y_i}))$$

关于w_{i0}的偏导数为

$$\frac{\partial \ln L}{\partial w_{i0}}=\sum_{k=1}^{N}(u_{ik}-\frac{\exp(w_i^T x_k+w_{i0})}{1+\sum_{i=1}^{c-1}\exp(w_i^T x_k+w_{i0})})$$

$$=\sum_{k=1}^{N}(u_{ik}-Sim_Y(x_k,\underline{Y_i}))$$

Newton-Raphson 算法求解

计算二阶导：

$$\frac{\partial^2 \ln L}{\partial w_i \partial w_j} = -\sum_{k=1}^{N} x_k x_k^T Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

$$\frac{\partial^2 \ln L}{\partial w_i \partial w_{j0}} = -\sum_{k=1}^{N} x_k Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

$$\frac{\partial^2 \ln L}{\partial w_{i0} \partial w_{j0}} = -\sum_{k=1}^{N} Sim_Y(x_k, \underline{Y_i}) Sim_Y(x_k, \underline{Y_j})$$

与softmax回归一样

Newton-Raphson 算法求解

Hessian矩阵H为$(c-1)\times(c-1)$矩阵

$$H=\begin{pmatrix} H_{11} & H_{11} & \cdots & H_{1(c-1)} \\ H_{21} & H_{22} & \cdots & H_{2(c-1)} \\ \cdots & \cdots & \cdots & \cdots \\ H_{(c-1)1} & H_{(c-1)2} & \cdots & H_{(c-1)(c-1)} \end{pmatrix} \quad H_{ij}=\begin{pmatrix} \dfrac{\partial^2 \ln L}{\partial w_{i0}\partial w_{j0}} & \left(\dfrac{\partial^2 \ln L}{\partial w_{i0}\partial w_j}\right)^T \\ \dfrac{\partial^2 \ln L}{\partial w_i \partial w_{j0}} & \dfrac{\partial^2 \ln L}{\partial w_i \partial w_j} \end{pmatrix}$$

H不可逆的概率大大降低

$$\beta=(\beta_1,\beta_2,\cdots,\beta_{c-1})^T \qquad \beta_i=(w_{i0},(w_i)_1,(w_i)_2,\cdots,(w_i)_p)$$

Logistic回归算法中，β的更新迭代公式为

$$\beta \leftarrow \beta - \mathrm{H}^{-1}\frac{\partial L}{\partial \beta}$$

对新的测试样本，通过类相似性映射函数$\mathrm{Sim}_{Y}(x,\underline{Y_i})$计算样本与每个类的相似度，选择其中最大的一个类作为所属类别。

Newton-Raphson 算法求解

Hessian矩阵H为$(c-1)\times(c-1)$矩阵

$$H=\begin{pmatrix} H_{11} & H_{11} & \cdots & H_{1(c-1)} \\ H_{21} & H_{22} & \cdots & H_{2(c-1)} \\ \cdots & \cdots & \cdots & \cdots \\ H_{(c-1)1} & H_{(c-1)2} & \cdots & H_{(c-1)(c-1)} \end{pmatrix} \quad H_{ij}=\begin{pmatrix} \frac{\partial^2 lnL}{\partial w_{i0}\partial w_{j0}} & \left(\frac{\partial^2 lnL}{\partial w_{i0}\partial w_j}\right)^T \\ \frac{\partial^2 lnL}{\partial w_i \partial w_{j0}} & \frac{\partial^2 lnL}{\partial w_i \partial w_j} \end{pmatrix}$$

H不可逆的概率大大降低

$$\beta=(\beta_1,\beta_2,\cdots,\beta_{c-1})^T \qquad \beta_i=(w_{i0},(w_i)_1,(w_i)_2,\cdots,(w_i)_p)$$

Logistic回归算法中，β的更新迭代公式为

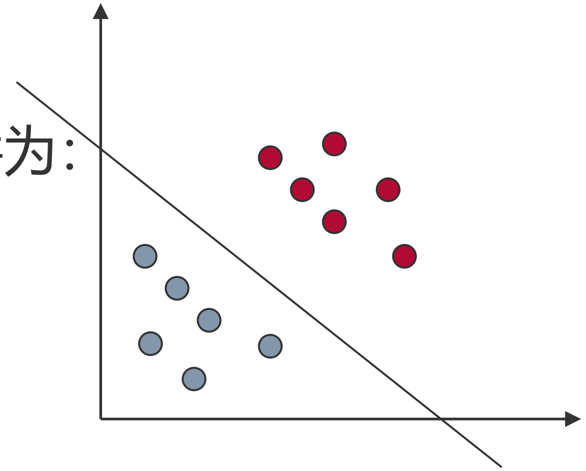
$$\beta \leftarrow \beta-\mathrm{H}^{-1}\frac{\partial L}{\partial \beta}$$

对新的测试样本，通过类相似性映射函数$\mathrm{Sim}_Y(x,\underline{Y}_i)$计算样本与每个类的相似度，选择其中最大的一个类作为所属类别。

Logistic回归--二分类

我们介绍Logistic回归如何处理二分类问题

图中数据利用线性函数 $\theta^T x = 0$ 分为两类，判别条件为：



$$y = g(\theta^T x) = \begin{cases} 1, & \theta^T x \geq 0 \\ 0, & \theta^T x < 0 \end{cases}$$

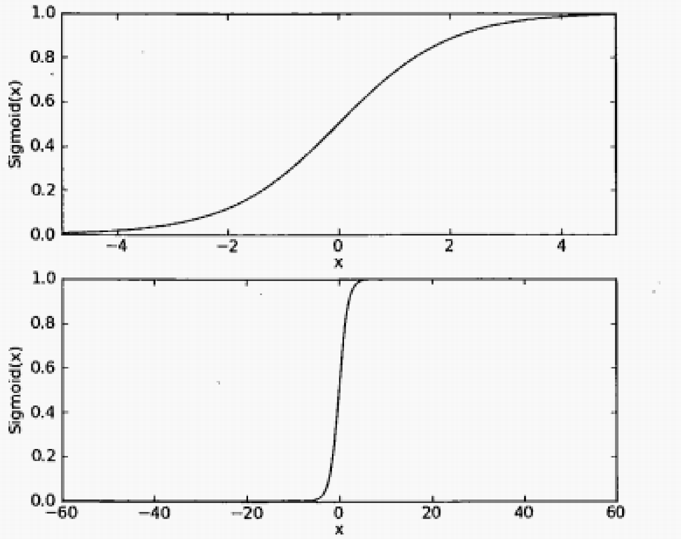
其中，$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix}$，$\mathrm{x}^{(i)} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}$，$i = 1, \cdots m$。

函数输出为0或1，引入Sigmoid函数：

$$g(z) = \frac{1}{1 + e^{-z}}$$

这样，得到分类预测函数：

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



两种坐标尺度下的Sigmoid函数图。上图的横坐标为−5到5，这时的曲线变化较为平滑；下图横坐标的尺度足够大，可以看到，在x＝0点处Sigmoid函数看起来很像阶跃函数

Logistic回归--二分类

其中，向量x是已知的，这样，求解分类函数问题转化为求向量θ：

$$P(y=1|x;\theta)=h_\theta(x)$$
$$P(y=0|x;\theta)=1-h_\theta(x)$$

合并可得

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

其中，$y\in\{0,1\}$。下面，利用最大似然函数对向量θ求解：

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

通过对数的形式对似然函数进行变换：

$$l(\theta)=\log L(\theta)$$
$$=\sum_{i=1}^{m}\left(y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right)$$

Logistic回归--二分类

接下来，利用梯度下降法求解θ，先构造损失函数：

$$Cast(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)), & y=1 \\ -\log(1-h_\theta(x)), & y=0 \end{cases}$$

代价函数 $J(\theta)$如下：

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

$$= -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

初始化 θ_j ： $$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta), j=0,1,2$$

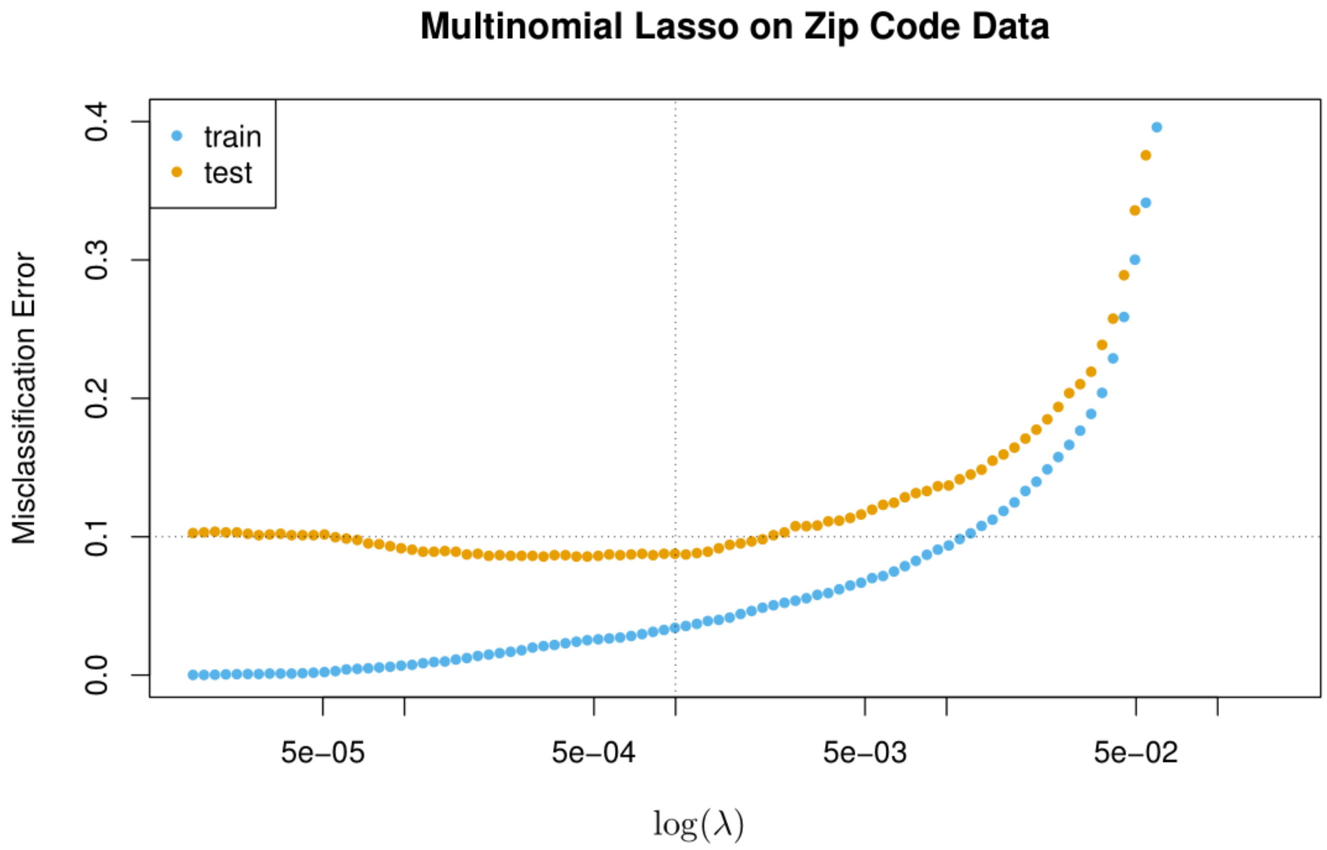
求偏导： $$\frac{\partial J(\theta)}{\partial\theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}, j=0,1,2$$

迭代公式为： $$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}, j=0,1,2$$

应用示例--手写体识别

美国邮局手写数字识别库（库中均为16×16像素的灰度图像，灰度值已归一化

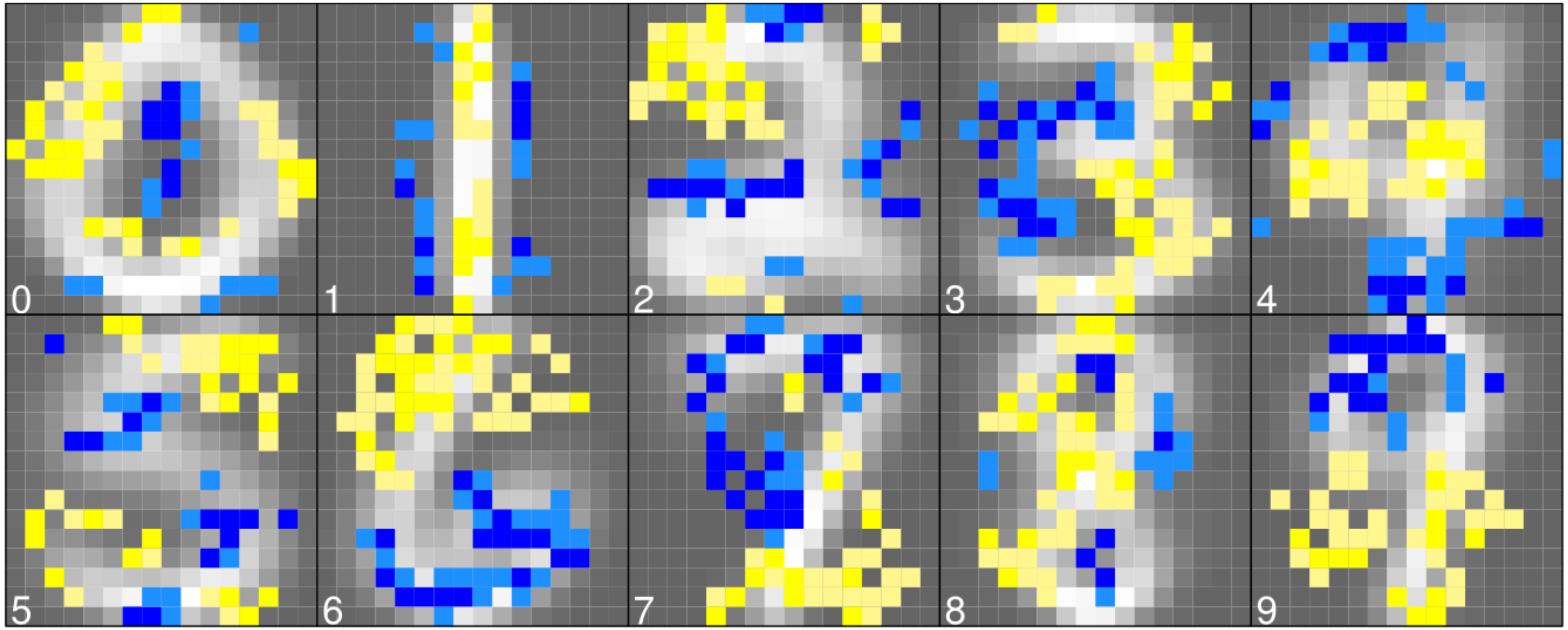
以p = 256个像素为特征点，拟合一个10类的Logistic回归Lasso模型



图中显示了用不同λ值对应的训练和测试误分类误差

应用示例--手写体识别

以图像的形式显示模型系数（平均约25%是非零系数）

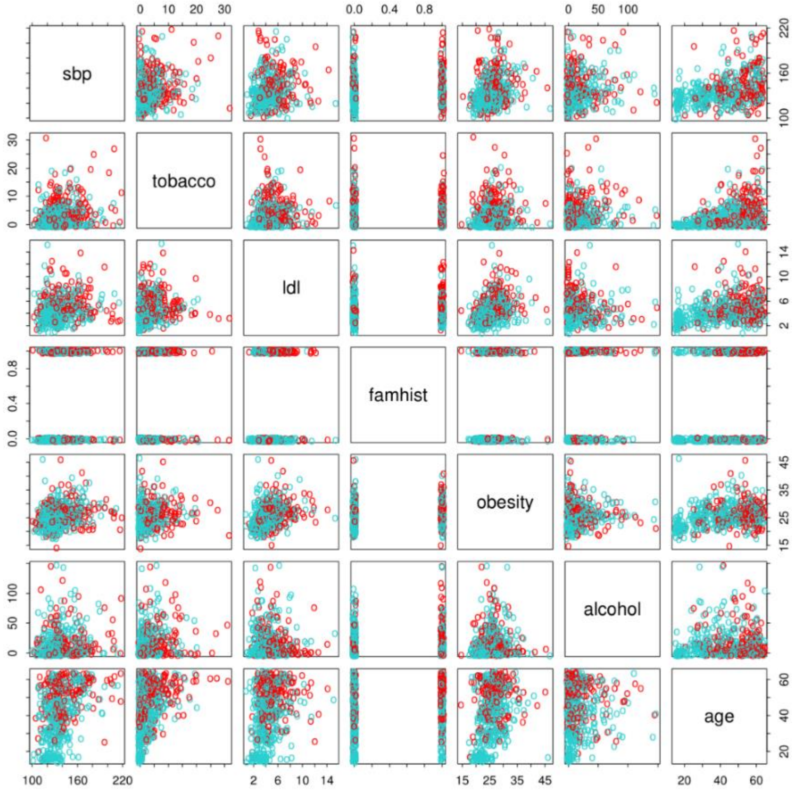


应用示例--南非心脏病问题

根据样本确定高发地区缺血性心脏病危险因素的强度

数据代表15 – 64岁的白人男性，反应变量是调查时是否存在心肌梗死

有160个案例，302个样本控制因子



南非心脏病数据的部分散点图矩阵（每个图显示一对危险因素，病例和对照用颜色编码(红色是病例)。可变的家族心脏病史(famhist)是二元的(是或不是)）

应用示例--南非心脏病问题

用最大似然拟合一个逻辑回归模型得到如下结果（不显著的Z分数表明可以从模型中去掉该系数，Z值绝对值大于约2是显著的）

	Coefficient	Std. Error	Z Score
(Intercept)	−4.130	0.964	−4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	−1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

- 收缩压(sbp)影响不显著?
- 肥胖(obesity)不显著?
- 多个因素互相关联的结果

模型选择：找出一个足够解释它们对冠心病患病率的共同影响的变量子集

- 去掉最不显著的系数，重新设计模型(重复，直到不再有任何项可从模型中删除)，得到模型如下

	Coefficient	Std. Error	Z score
(Intercept)	−4.204	0.498	−8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

- 删除一个变量重新构建每个模型，然后进行偏差分析，以决定排除哪个变量(更耗时)

总结

SoftMax

类认知表示

$$\forall i, \underline{Y_i} = (x, w_{i0} + w_i^T x)$$

类相似性映射

$$\forall k \forall i \quad \frac{\exp(w_{i0}+w_i^T x_k)}{\sum_{j=1}^{c}\exp(w_{j0}+w_j^T x_k)}$$

目标函数

$$\log L = \sum^{N} \log Sim_Y(x_k, \underline{Y_{\overrightarrow{x_k}}}) = \sum^{N}\sum^{c} u_{ik}\log Sim_Y(x_k, \underline{Y_i})$$
$$= \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}\left(w_{i0} + w_i^T x_k - \log\sum_{j=1}^{c}\exp(w_{j0} + w_j^T x_k)\right)$$

Logistic

$$1 \le i \le c-1 \quad \underline{Y_i} = (x, w_i^T x + w_{i0})$$

$$i = c \quad \underline{Y_c} = (x, F_c(x)), F_c(x)\text{未知}$$

$$1 \le i \le c-1 \quad \frac{\exp({w_i}^T x + w_{i0})}{1+\sum_{i=1}^{c-1}\exp({w_i}^T x + w_{i0})}$$

$$i = c \quad \frac{1}{1+\sum_{i=1}^{c-1}\exp({w_i}^T x + w_{i0})}$$

$$\max_{\underline{Y_1},\underline{Y_2},\ldots,\underline{Y_{c-1}}} \log L = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}\log Sim_Y(x_k, \underline{Y_i})$$
$$= \sum_{k=1}^{N}\sum_{i=1}^{c-1} u_{ik}(w_i x_k^T + w_{i0}) - \sum_{k=1}^{N}\log\left(1 + \sum_{i=1}^{c-1}\exp(w_i x_k^T + w_{i0})\right)$$

总结

- 类认知表示有区别：
  - 在softmax回归中，每一类的类认知表示是确定的
  - 在logistic回归中，c-1类的类认知表示是确定的，第c类未知，可以是任何形式
  - 第c类的认知表示对logistic没有影响， logistic回归又称为对数几率回归。
- Softmax适用于互斥的分类问题，logistic适用于非互斥问题
- 联系：从标准softmax回归分类模型消除多余参数，可以导出logistic回归分类模型。
- 对数线性分类模型的两个重要特点
  - 类认知表示是确定性的，不含不确定因素
  - 类相似函数可以用伪后验概率密度表示，具有一定的不确定信息
- 对数线性分类模型处于确定性分类模型和概率分类模型交界处。
- 属于广义线性模型：自变量的线性预测的函数是因变量的估计值。

总结

- 设计一个算法，除了类认知表示，类相似性映射也很重要。

- 在类认知表示相同的情况下，不同的类相似性映射可以导出完全不同的分类算法。

- 设计算法容易，现实合理性不容易。理论合理性针对的是理想情况，需要增加现实约束，才能设计一个实用的模型。

练习题

1. 证明：点$x \in \mathbb{R}^n$到超平面$a^t x + b = 0$的距离为$\frac{|a^t x + b|}{\|a\|_2}$。

2. 考虑Logistic回归的对数似然函数

$$l(\theta) = \sum_{i=1}^{m} y_i \log h(x_i) + (1 - y_i) \log(1 - h(x_i))$$

其中，$h(x) = \varphi(\theta^t x)$，$\varphi(z) = \frac{1}{1+e^{-z}}$，$x_i \in \mathbb{R}^n$以及$y_i \in \mathbb{R}$。

   (a) 证明$\varphi'(z) = \varphi(z)(1 - \varphi(z))$以及$\frac{\partial h(x)}{\partial \theta_k} = h(x)(1 - h(x))x_k$。

   (b) 计算$l(\theta)$的Hessian矩阵H并证明该矩阵半负定，即对任何$x \in \mathbb{R}^n$都有$x^t H x \leq 0$。

   (c) 证明$l(\theta)$是一个凹函数。

3. 给定数据集$\{(a_i, b_i) : 1 \leq i \leq N\}$，其中$a_i \in \mathbb{R}^n$，$b_i \in \{\pm 1\}$，并存在线性函数$f(a) = x^t a + y$使得$b_i f(a_i) > 0$对任何$1 \leq i \leq N$成立。此时，利用MLE得到的最佳模型参数$(x, y)$是什么？

北京交通大学《机器学习》课程组成员

于　剑：jianyu@bjtu.edu.cn，http://faculty.bjtu.edu.cn/6463/

景丽萍：lpjing@bjtu.edu.cn，http://faculty.bjtu.edu.cn/8249/

田丽霞：lxtian@bjtu.edu.cn，http://faculty.bjtu.edu.cn/7954/

黄惠芳：hfhuang@bjtu.edu.cn，http://faculty.bjtu.edu.cn/7418/

杨　凤：fengyang@bjtu.edu.cn，http://faculty.bjtu.edu.cn/8518/

吴　丹：wudan@bjtu.edu.cn，http://faculty.bjtu.edu.cn/8925/

万怀宇：hywan@bjtu.edu.cn，http://faculty.bjtu.edu.cn/8793/

王　晶：wj@bjtu.edu.cn，http://faculty.bjtu.edu.cn/9167/