机器学习：监督学习

机器学习基本概念

loss function

- 0-1 loss
- 平方损失
- 绝对损失
- 对数损失/对数似然损失：$Loss(y_i, P(y_i|x_i)) = -\log(P(y_i|x_i))$

- 经验风险：越小，说明学习模型对训练数据拟合程度越好
- 期望风险：期望风险越小，学习所得模型越好

结构风险最小化(structural risk minimization)

为了防止过拟合，在经验风险上加上表示模型复杂度的正则化项或惩罚项

判别方法直接学习判别函数f(X)或者条件概率分布P(Y|X)作为预测的模型，即判别模型

生成模型从数据中学习联合概率分布P(X, Y)

回归分析

一元线性回归的推导：让loss对b的偏导为0，求出b的表达，再带回loss中，对a求偏导，求出a的表达

多元线性回归：核心是变成矩阵来看

$$J_m(a) = (y - X^T a)^T (y - X^T a)$$

对参数a求导：$\nabla J(\boldsymbol{a}) = -2X(y - X^T\boldsymbol{a})$

令上式为0：

$$XX^T a = Xy$$

$$a = (XX^T)^{-1} Xy$$

线性回归的问题：对离群点非常敏感，导致模型建模不稳定，使结果有偏

解决方法：logistic regression

回归模型中引入sigmoid函数的非线性回归模型

$$y = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{x}+b)}} \quad ，其中 y \in (0,1), z = \boldsymbol{w}^T\boldsymbol{x} + \mathrm{b}$$

上式的输出为[0,1]，非常适合二分类问题：

p(y=1|x)　　和　　p(y=0|x) = 1 - p(y=1|x)

我们现在对比值$\frac{p}{1-p}$取对数(即$\log\left(\frac{p}{1-p}\right)$)来表示输入数据$\boldsymbol{x}$属于正例概率。$\frac{p}{1-p}$被称为几率(odds)，反映了输入数据$\boldsymbol{x}$作为正例的相对可能性。$\frac{p}{1-p}$的对数几率(log odds)或logit函数可表示为$\log\left(\frac{p}{1-p}\right)$。

显然，可以得到$p(y=1|\boldsymbol{x}) = h_\theta(x) = \frac{1}{1+e^{-(w^T x+b)}}$和$p(y=0|\boldsymbol{x}) = 1 - h_\theta(x) = \frac{e^{-(w^T x+b)}}{1+e^{-(w^T x+b)}}$。$\theta$表示模型参数（$\theta = \{\boldsymbol{w}, b\}$）。于是有：

$$\operatorname{logit}(p(y=1|\boldsymbol{x})) = \log\left(\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}\right) = \log\left(\frac{p}{1-p}\right) = \boldsymbol{w}^T\boldsymbol{x} + \mathrm{b}$$

输入数据x属于正例的概率大于其属于负例的概率，即p(y=1 | x) > 0.5，则被判为正例，也就是$w^T x + b > 0$成立

logistic回归本质上是一个线性模型，预测时可以计算线性函数取值是否大于0来判断输入数据x的类别归属

模型参数的似然函数被定义为$\mathcal{L}(\theta|\mathcal{D}) = \mathrm{p}(\mathcal{D}|\theta)$，其中$\mathcal{D} = \{(x_i, y_i\}(1 \le \mathrm{i} \le \mathrm{n})$表示所有观测数据（或训练数据），$\theta$表示模型参数（$\theta = \{\boldsymbol{w}, b\}$）。在最大化对数似然函数过程中，一般假设观测所得每一个样本数据是独立同分布(independent and identically distributed, i.i.d），于是可得：

$$\mathcal{L}(\theta|\mathcal{D}) = \mathrm{p}(\mathcal{D}|\theta) = \prod_{i=1}^{n} \mathrm{p}(y_i|\mathrm{x}, \theta) = \prod_{i=1}^{n} \left(h_\theta(x_i)\right)^{y_i}\left(1 - h_\theta(x_i)\right)^{1-y_i}$$

$$p(y=1|\boldsymbol{x}) = h_\theta(x) = \frac{1}{1+e^{-(w^T x+b)}}$$

对上述公式取对数：

$$\mathrm{l}(\theta) = \log(L(\theta|\mathcal{D})) = \sum_{i=1}^{n} y_i \log(h_\theta(x_i)) + (1-y_i)\log(1-h_\theta(x_i))$$

最大似然估计目的是计算似然函数的最大值，而分类过程是需要损失函数最小化。因此，在上式前加一个负号得到损失函数(cross entropy, 交叉熵)：

$$\begin{aligned}\mathcal{T}(\theta) &= -\mathrm{l}(\theta) = -\log(L(\theta|\mathcal{D})) \\ &= -(\sum_{i=1}^{n} y_i \log(h_\theta(x_i)) + (1-y_i)\log(1-h_\theta(x_i)))\end{aligned}$$

$\mathcal{T}(\theta)$等价于：$\mathcal{T}(\theta) = \begin{cases} -\log(h_\theta(x_i)) & if\ y=1 \\ -\log(1-h_\theta(x_i)) & if\ y=0 \end{cases}$

> 6.16:
>
> log累加，其实就是各个样本的条件概率都加和

求总的损失函数$J(\theta)$时，利用y = 1或者0的情况，直接在前面乘y或者(1-y)

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= -\sum_{i=1}^{n}\left(y_i \frac{1}{h_\theta(x_i)}\frac{\partial h_\theta(x_i)}{\partial \theta_j} + (1-y_i)\frac{1}{1-h_\theta(x_i)}\frac{\partial(1-h_\theta(x_i))}{\partial \theta_j}\right) \\
&= -\sum_{i=1}^{n}\left(\frac{y_i}{h_\theta(x_i)} - \frac{1-y_i}{1-h_\theta(x_i)}\right)\frac{\partial h_\theta(x_i)}{\partial \theta_j} \\
&= -\sum_{i=1}^{n} x_i h_\theta(x_i)(1-h_\theta(x_i))\left(\frac{y_i}{h_\theta(x_i)} - \frac{1-y_i}{1-h_\theta(x_i)}\right) \\
&= -\sum_{i=1}^{n} x_i(y_i(1-h_\theta(x_i)) - (1-y_i)h_\theta(x_i)) \\
&= \sum_{i=1}^{n}(h_\theta(x_i) - y_i)x_i
\end{aligned}
$$

将求导结果代入梯度下降迭代公式得:

$$\theta_j = \theta_j - \eta\sum_{i=1}^{n}(h_\theta(x_i) - y_i)x_i$$

决策树

entropy

假设有K个信息（类别），其组成了集合样本D，记第k个信息（类别）发生的概率为$p_k(1 \le k \le K)$。如下定义这K个信息的信息熵：

$$Ent(D) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

$Ent(D)$值越小，表示D包含的信息越确定，也称D的纯度越高。需要指出，所有p_k累加起来的和为1。

计算信息熵时约定：若p_k=0，则$p_k \log_2 p_k$=0

信息增益:

$$Gain(D,A) = Ent(D) - \sum_{i=1}^{n}\frac{|D_i|}{|D|}Ent(D_i)$$

对**可取值数目较多**的属性有所偏好

增益率:

$info$和$Gain-ratio$（增益率）计算公式如下：

$$info = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$ 信息熵

$$Gain-ratio = Gain(D,A)/info$$

对**可取值数目较少的属性**有所偏好

> 6.16：
>
> info的含义：划分时产生的类的信息
>
> 分类多, info大，增益率小

C4.5:先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的

CART采用基尼指数来选择划分属性：

$$\mathrm{Gini(D)} = 1 - \sum_{k=1}^{K} p_k^2$$

连续属性离散化（二分法）

对于已有的取值，计算相邻两点之间的均值，作为候选划分点

> 6.16：
>
> 然后对于这些均值，计算Gain，Gain越大，划分点越好

剪枝处理

- 避免过拟合
  - 预剪枝：避免过拟合，但有可能欠拟合
  - 后剪枝：欠拟合风险小，泛化性能一般优于预剪枝，但是训练时间开销大
- 判断决策树泛化性能是否提升的方法：
  - 留出法：预留一部分数据用作“验证集”以进行性能评估

线性区别分析

- 基于监督学习的降维方法
- 类内方差小，类间间隔大

定义$\boldsymbol{X}$为所有样本构成集合、N_i为第i个类别所包含样本个数、X_i为第i类样本的集合、$\boldsymbol{m}$为所有样本的**均值向量**、$\boldsymbol{m}_i$为第i类样本的**均值向量**。Σ_i为第i类样本的协方差矩阵，其定义为:

$$\Sigma_i = \sum_{x \in \boldsymbol{X}_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

投影之后类别C_1的协方差矩阵s_1为:

$$s_1 = \sum_{x \in C_1} (\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{m}_1)^2 = \boldsymbol{w}^T \sum_{x \in C_1} [(\boldsymbol{x} - \boldsymbol{m}_1)\,(\boldsymbol{x} - \boldsymbol{m}_1)^T]\boldsymbol{w}$$

同理可得到投影之后类别C_2的协方差矩阵s_2。

$$\Sigma_i = \sum_{x \in \boldsymbol{X}_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

$s1 = w^T \sum_1 w$

$s2 = w^T \sum_2 w$

类内方差小：s1+s2

在投影之后的空间中，归属于两个类别的数据样本中心可分别如下计算:

$$m_1 = \boldsymbol{w}^T\boldsymbol{m}_1,\quad m_2 = \boldsymbol{w}^T\boldsymbol{m}_2$$

类间间隔大：最大化$||m_2 - m_1||_2^2$

结合以上两点，最大化下式:

$$J(\boldsymbol{w}) = \frac{\|m_2 - m_1\|_2^2}{s_1 + s_2}$$

$$J(\boldsymbol{w})=\frac{\|\boldsymbol{w}^T(\boldsymbol{m}_2-\boldsymbol{m}_1)\|_2^2}{\boldsymbol{w}^T\Sigma_1\boldsymbol{w}+\boldsymbol{w}^T\Sigma_2\boldsymbol{w}}=\frac{\boldsymbol{w}^T(\boldsymbol{m}_2-\boldsymbol{m}_1)(\boldsymbol{m}_2-\boldsymbol{m}_1)^T\boldsymbol{w}}{\boldsymbol{w}^T(\Sigma_1+\Sigma_2)\boldsymbol{w}}=\frac{\boldsymbol{w}^T\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}}$$

其中，$\boldsymbol{S}_b$称为类间散度矩阵(between-class scatter matrix)，即衡量两个类别均值点之间的“分离”程度，可定义如下：

$$\boldsymbol{S}_b=(\boldsymbol{m}_2-\boldsymbol{m}_1)(\boldsymbol{m}_2-\boldsymbol{m}_1)^T$$

$\boldsymbol{S}_w$则称为类内散度矩阵(within-class scatter matrix)，即衡量每个类别中数据点的“分离”程度，可定义如下：

$$\boldsymbol{S}_w=\Sigma_1+\Sigma_2$$

可以让分母等于1，问题变成了最大化分子，用拉格朗日函数求解

对应拉格朗日函数为：

$$L(\boldsymbol{w})=\boldsymbol{w}^T\boldsymbol{S}_b\boldsymbol{w}-\lambda(\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}-1)$$

$$2S_b\mathbf{w}=2\lambda S_w\mathbf{w}. \Rightarrow S_W^{-1}S_B\boldsymbol{w}=\lambda\boldsymbol{w}$$

对$\boldsymbol{w}$求偏导并使其求导结果为零，可得$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w}=\lambda\boldsymbol{w}$。由此可见，$\lambda$和$\boldsymbol{w}$分别是$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b$的特征根和特征向量，$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w}=\lambda\boldsymbol{w}$也被称为Fisher线性判别（Fisher linear discrimination）。

因为$\boldsymbol{S}_b\boldsymbol{w}=(\boldsymbol{m}_2-\boldsymbol{m}_1)(\boldsymbol{m}_2-\boldsymbol{m}_1)^T\boldsymbol{w}$，其中$(\boldsymbol{m}_2-\boldsymbol{m}_1)^T\boldsymbol{w}$是个实数，不妨令$(\boldsymbol{m}_2-\boldsymbol{m}_1)^T\boldsymbol{w}=\lambda_w$，则：

$$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{w}=\boldsymbol{S}_w{}^{-1}(\boldsymbol{m}_2-\boldsymbol{m}_1)\times\lambda_w=\lambda\boldsymbol{w}$$

由于对$\boldsymbol{w}$的放大和缩小操作不影响结果，因此可约去上式中的未知数λ和λ_w，得到：$\boldsymbol{w}=\boldsymbol{S}_w{}^{-1}(\boldsymbol{m}_2-\boldsymbol{m}_1)$ $\boxed{\mathbf{w}\propto\mathbf{S}_\mathbf{W}^{-1}(\mathbf{m}_2-\mathbf{m}_1)}$

这里m和w都是n*1的。核心在于求出fisher线性判别后，根据定义带入S_b，然后把最后的化为一个实数进行约简，最后w只和S_w^{-1}, m2, m1有关

> 分别求出投影数据的均值和方差，就可以求出最佳投影方向w

线性区别分析：多分类问题

假设n个原始高维数据所构成的类别种类为K、每个原始数据被投影映射到低维空间中的维度为r。

令投影矩阵$\boldsymbol{W}=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_r)$，可知$\boldsymbol{W}$是一个$n\times r$矩阵。于是，$\boldsymbol{W}^T\boldsymbol{m}_i$为第$i$类样本数据中心在低维空间的投影结果，$\boldsymbol{W}^T\Sigma_i\boldsymbol{W}$为第$i$类样本数据协方差在低维空间的投影结果。

“类内方差小、类间间隔大”

类内散度矩阵$\boldsymbol{S}_w$重新定义如下：

$$\boldsymbol{S}_w=\sum_{i=1}^{K}\Sigma_i,\ \text{其中}\Sigma_i=\sum_{x\in class\,i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^T$$

在上式中，$\boldsymbol{m}_i$是第i个类别中所包含样本数据的均值。

类间散度矩阵$\boldsymbol{S}_b$重新定义如下：

$$\boldsymbol{S}_b=\sum_{i=1}^{K}\frac{N_i}{N}(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^T$$

w的含义发生了变化，从原来的n*1变成了n*r矩阵

类内散度矩阵S_w仍然是各个类的所有样本和该类均值的差的平方和

类间散度矩阵只是计算各个类和整体均值差，但是前面加了该类占总的比例

将多类LDA映射投影方向的优化目标$J(\boldsymbol{W})$改为：

$$J(\boldsymbol{W})=\frac{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_b\boldsymbol{W}}{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_w\boldsymbol{W}}$$

其中，$\prod_{diag}\mathbb{A}$为矩阵$\mathbb{A}$主对角元素的乘积。

继续对$J(\boldsymbol{W})$进行变形：

$$J(\boldsymbol{W})=\frac{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_b\boldsymbol{W}}{\prod_{diag}\boldsymbol{W}^T\boldsymbol{S}_w\boldsymbol{W}}=\frac{\prod_{i=1}^{r}\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\prod_{i=1}^{r}\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}=\prod_{i=1}^{r}\frac{\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}$$

显然需要使乘积式子中每个$\frac{\boldsymbol{w}_i^T\boldsymbol{S}_b\boldsymbol{w}_i}{\boldsymbol{w}_i^T\boldsymbol{S}_w\boldsymbol{w}_i}$取值最大，这就是二分类问题的求解目标，即每一个$\boldsymbol{w}_i$都是$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_b\boldsymbol{W}=\lambda\boldsymbol{W}$的一个解。

这里的S_w,S_b都是n*n的，和之前的一样，W是n*r的

最后能推导出主对角线上的矩阵都是形如$w_i^TS_bw_i$，i∈[1, r]

对线性判别分析的降维步骤描述如下：

1. 计算数据样本集中每个类别样本的均值

2. 计算类内散度矩阵S_w和类间散度矩阵S_b
3. 根据$S_w^{-1}S_bW=\lambda W$来求解$S_w^{-1}S_b$所对应前r个最大特征值对应特征向量(w1, w2, ..., w_r)，构成矩阵W
4. 通过矩阵W将每个样本映射到低维空间，实现特征降维

上述第三步和二分类的不一样，这里是因为要做出r维，所以选前r个最大特征值对应特征向量，二分类的只是直接计算了

> LDA降维依赖于标签，也就是初始人为分好的K个类，但是PCA是无监督的，直接计算数据的协方差然后取特征值，和有多少个类没关系

Ada Boosting

霍夫丁不等式

$$P(|x-y|\geq\epsilon)\leq 2e^{-2N\epsilon^2}(N\text{是采样人口总数、}\epsilon\in(0,1)\text{是所设定的可容忍误差范围})$$

ada boosting两个核心问题:

- 如何改变训练数据的权重
- 如何将一系列弱分类器组合成强分类器：提高分类误差小的弱分类器的权重，减少分类误差大的弱分类器的权重

计算第m个弱分类器$G_m(x)$的权重α_m: $\alpha_m=\frac{1}{2}ln\frac{1-err_m}{err_m}$

更新训练样本数据的分布权重:

更新训练样本数据的分布权重： $D_{m+1}=(w_{m+1,1},\dots,w_{m+1,i},\dots,w_{m+1,N}),w_{m+1,i}=\frac{w_{m,i}}{Z_m}e^{-\alpha_m y_i G_m(x_i)}$，其中$Z_m$是归一化因子以使得$D_{m+1}$为概率分布，$Z_m=\sum_{i=1}^{N}w_{m,i}\,e^{-\alpha_m y_i G_m(x_i)}$

$$w_{m+1,i}=\begin{cases}\frac{w_{m,i}}{Z_m}e^{-\alpha_m}, & G_m(x_i)=y_i\\ \frac{w_{m,i}}{Z_m}e^{\alpha_m}, & G_m(x_i)\neq y_i\end{cases}$$

$$w_{m+1,i}=\begin{cases}\frac{w_{m,i}}{Z_m}e^{-\alpha_m}, & G_m(x_i)=y_i\\ \frac{w_{m,i}}{Z_m}e^{\alpha_m}, & G_m(x_i)\neq y_i\end{cases}$$