机器学习
Machine Learning

第四讲：分类的基本学习算法

注：

本章讲义符号与前章略有不同，

用t表示标注，y表示模型输出

本讲义采用两套符号之一：

用y表示标注，$\hat{y}$ 表示模型输出

或：

用t表示标注，y表示模型输出

1.分类问题（Classification）

- 数据集（标注集）

$$\mathbb{D} = \left\{ \left(\boldsymbol{x}_i, \boldsymbol{t}_i \right) \right\}_{i=1}^{N} \quad \Rightarrow \boldsymbol{y} = h\left(\boldsymbol{x} \right)$$

- 分类(Classification) $t, y \in \{1, \cdots, \mathrm{C}\}$

$$\mathrm{C} = 2, \quad \textit{binary} \quad \textit{classification}$$

$$C > 2, \text{multiclass} \quad \textit{classification}$$

基本分类问题表示

对于2类问题

$$t \in \{0,1\}$$

$$t = 1 \Rightarrow C1 \quad t = 0 \Rightarrow C2$$

对于K (>2)类问题（K-to-1编码）

$$\boldsymbol{t} = \left(0, \cdots, 0, 1, 0, \cdots, 0\right)^T$$

分类的三种基本模型

(1). 判决函数模型

线性模型

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

广义线性模型（generalized linear models）

$$y(\boldsymbol{x}) = f\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right)$$

$f(\ )$ 为激活函数

<u>由训练集学习参数 $\boldsymbol{w}, w_0$ （或用验证集确定超参数）</u>

$\mathbf{X}$ 表示数据集

分类的三种基本模型（续）

(2). 判决（概率）模型

由数据集直接训练后验概率，由决策论确定输出

$$p\left(C_k \middle| \mathbf{x};\mathbf{X}\right) \quad \text{简写为} \quad p\left(C_k \middle| \mathbf{x}\right)$$

分类的三种基本模型（续）

(3). 生成（概率）模型

首先得到 $p\left(\boldsymbol{x} \middle| C_k\right)$ 和 $p\left(C_k\right)$ 或：$p\left(\boldsymbol{x}, C_k\right)$

再由Bayes公式得

$$p\left(C_k \middle| \boldsymbol{x}\right) = \frac{p\left(\boldsymbol{x} \middle| C_k\right) p\left(C_k\right)}{p\left(\boldsymbol{x}\right)}$$

$p\left(\boldsymbol{x}\right)$ $p\left(\boldsymbol{x}, C_k\right)$ 可以求得，能够生成更多数据

2.判决函数方法（Discriminant Functions）

- 线性分类
  - （2类、多类），LS优化
- Fisher线性判决函数
  - （2类、多类）（统计方法的传统技术）
  - 投影到低维空间，投影可分辨力最大化
- 感知器算法（The Perceptron Algorithm）
  - （MLP的最简化形式，曾起到重要作用）
- 传统算法（略，祥见教材4.2节）

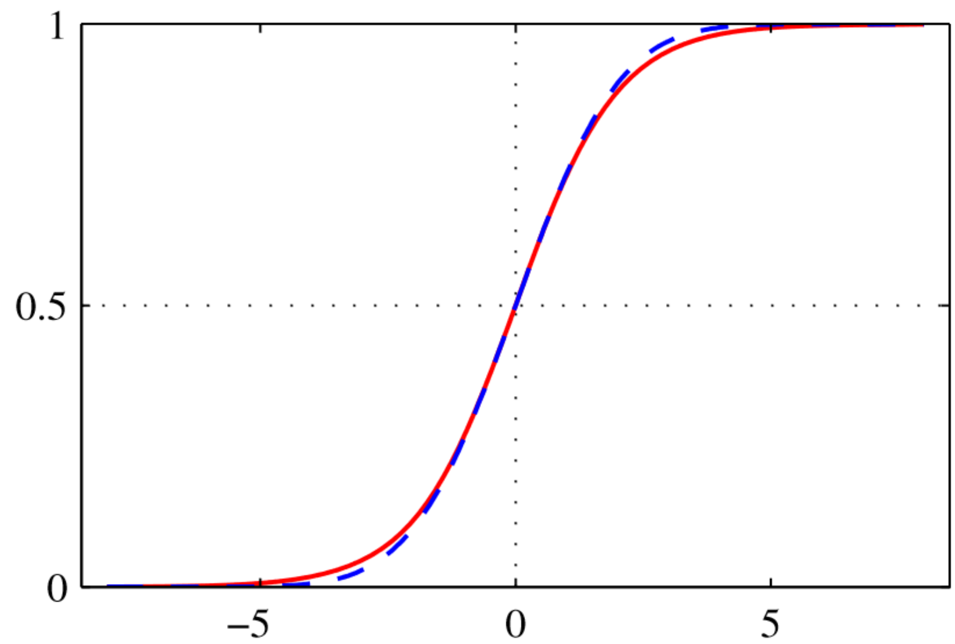
3.逻辑回归（Logistic Regression）（概率判决模型）

3.1二类问题

用logistic sigmoid表示类的后验概率

定义：logistics sigmoid函数

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$



本节直接采用基函数$\varphi(x)$
线性形式时取$\varphi(x)=\bar{x}$

logistics sigmoid函数的性质

$$\sigma(-a) = 1 - \sigma(a)$$

$$\frac{d\sigma(a)}{da} = \sigma(a)(1 - \sigma(a))$$

对权向量的线性系统

$$a = \boldsymbol{w}^T \phi(\boldsymbol{x})$$

其中

$$\boldsymbol{w} = [w_0, w_1, \cdots, w_{M-1}]^T$$

$$\phi(\boldsymbol{x}) = [1, \phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_{M-1}(\boldsymbol{x})]^T$$

二分类后验概率的表示

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right)$$

另：

$$p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi)$$

由训练样本直接学习参数 $\boldsymbol{w}$

对于新的输入 $\boldsymbol{x} \Rightarrow \phi(\boldsymbol{x})$

计算类后验概率$P\left(C_1\middle|\phi\right) = y(\phi) = \sigma\left(\boldsymbol{w}^T\phi(\boldsymbol{x})\right)$ 并分类

二分类逻辑回归参数学习

训练样本集 $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$

变换的样本集 $\{\phi(\mathbf{x}_n), t_n\}_{n=1}^{N} = \{\phi_n, t_n\}_{n=1}^{N}$

几个简写

$$\phi_n = \phi(\mathbf{x}_n)$$

$$\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$$

$$y_n = p(\mathcal{C}_1|\phi_n) \qquad y_n = \sigma(a_n)$$

$$a_n = \mathbf{w}^{\mathrm{T}}\phi_n$$

二分类逻辑回归参数学习（续）

似然函数

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

负对数似然函数

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w})$$

$$= -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

注意

$$y_n = \sigma(a_n)$$

含参数 $\boldsymbol{w}$

二分类逻辑回归参数学习（续）

目标函数对 $\boldsymbol{w}$ 的梯度，可导出为

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$$

$$= \sum_{n=1}^{N} \left(\sigma\left(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) \right) - t_n \right) \phi(\boldsymbol{x}_n)$$

<u>随机梯度</u>

$$\nabla E_n = (y_n - t_n)\phi_n$$

二分类逻辑回归参数学习（续）

随机梯度算法学习参数

$$
\begin{aligned}
\boldsymbol{w}^{(\mathrm{k}+1)} &= \boldsymbol{w}^{(\mathrm{k})} - \eta \nabla E_n \\
&= \boldsymbol{w}^{(\mathrm{k})} - \eta \left(y_n - t_n \right) \phi_n \\
&= \boldsymbol{w}^{(\mathrm{k})} - \eta \left(\sigma \left(\boldsymbol{w}^{(\mathrm{k})^T} \phi(\boldsymbol{x}_n) \right) - t_n \right) \phi(\boldsymbol{x}_n)
\end{aligned}
$$

η 学习率参数

注：可按一定次序使用 $\{\boldsymbol{x}_n, t_n\}_{n=1}^{N}$ ，甚至可循环使用直到收敛，也可用小批量平均梯度。

二分类逻辑回归参数学习（续）

IRLS算法*

（Iterative Reweighted Least Squares)

利用牛顿迭代思想

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1} \nabla E(\mathbf{w})$$

其中

$$\mathbf{H} = \nabla \nabla E(\mathbf{w}) \quad \text{Hessian matrix}$$

$\mathbf{\Phi}$ **NXM矩阵**

ϕ_n^{T} **第n行**

IRLS算法

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \mathbf{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t})$$

$$\begin{aligned}\mathbf{H} &= \nabla\nabla E(\mathbf{w}) \\ &= \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi}\end{aligned}$$

其中 $N \times N$ diagonal matrix $\mathbf{R}$ $\quad R_{nn} = y_n(1-y_n)$

H正定，有唯一最优解

IRLS算法

$$
\begin{aligned}
\mathbf{w}^{(\text{new})} &= \\
&= \mathbf{w}^{(\text{old})} - (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t}) \\
&= (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\left\{\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}\mathbf{w}^{(\text{old})} - \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y}-\mathbf{t})\right\} \\
&= (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{z}
\end{aligned}
$$

这里 $\mathbf{z} = \boldsymbol{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y}-\mathbf{t})$ N维向量

这是一个**R**作为加权矩阵的加权LS，每次需重新计算**R**，重新运行LS，直到收敛。故名：IRLS。

抗逻辑回归的overfitting!

正则化逻辑回归
（Regularized Logistic Regression）

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$= -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

$$+ \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

也可加更一般的正则化项，例如：

$$\frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

q=1（或<1）对应稀疏逻辑回归。

3.2 多类逻辑回归
Multiclass logistic regression

定义Softmax 函数

$$p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

这里 $a_k = \mathbf{w}_k^{\mathrm{T}}\phi$

对每一类定义类后验概率

对每一类定义和学习权向量 $\boldsymbol{w}_k$

多类逻辑回归
Multiclass logistic regression

Softmax 函数的性质

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$$

这里

$$I_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}$$

多类逻辑回归参数学习

训练样本集 $\left\{ \boldsymbol{x}_n, \boldsymbol{t}_n \right\}_{n=1}^{N}$

这里
$$\boldsymbol{t}_n = \left[0, \cdots, 0, 1, 0, \cdots, 0 \right]^T$$
$$= \left[t_{n1}, t_{n2}, \cdots, t_{nK} \right]^T$$

变换的样本集 $\left\{ \phi(\boldsymbol{x}_n), \boldsymbol{t}_n \right\}_{n=1}^{N} = \left\{ \phi_n, \boldsymbol{t}_n \right\}_{n=1}^{N}$

令： $\mathbf{T} = \left[t_{nk} \right]_{N \times K}$ $\quad y_{nk} = y_k(\phi_n)$

多类逻辑回归参数学习

似然函数

$$p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) =$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

多类逻辑回归参数学习

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K)$$
$$= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$$

存在约束条件

$$\sum_k t_{nk} = 1$$

多类逻辑回归参数学习

目标函数对各参数向量的梯度

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \, \phi_n$$

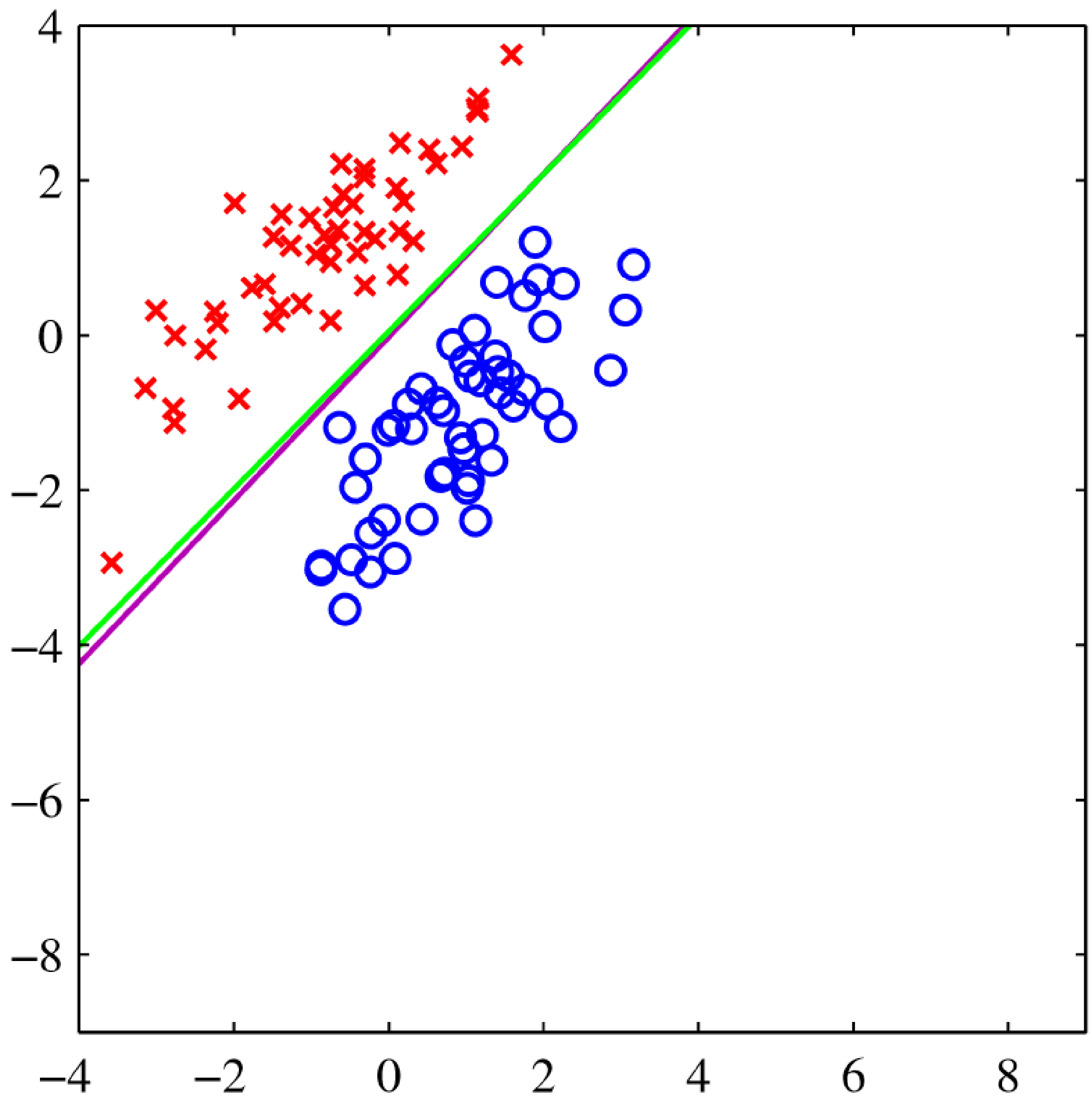
随机梯度

$$\nabla_{\mathbf{w_j}} E_n = \left(y_{nj} - t_{nj} \right) \phi_n$$

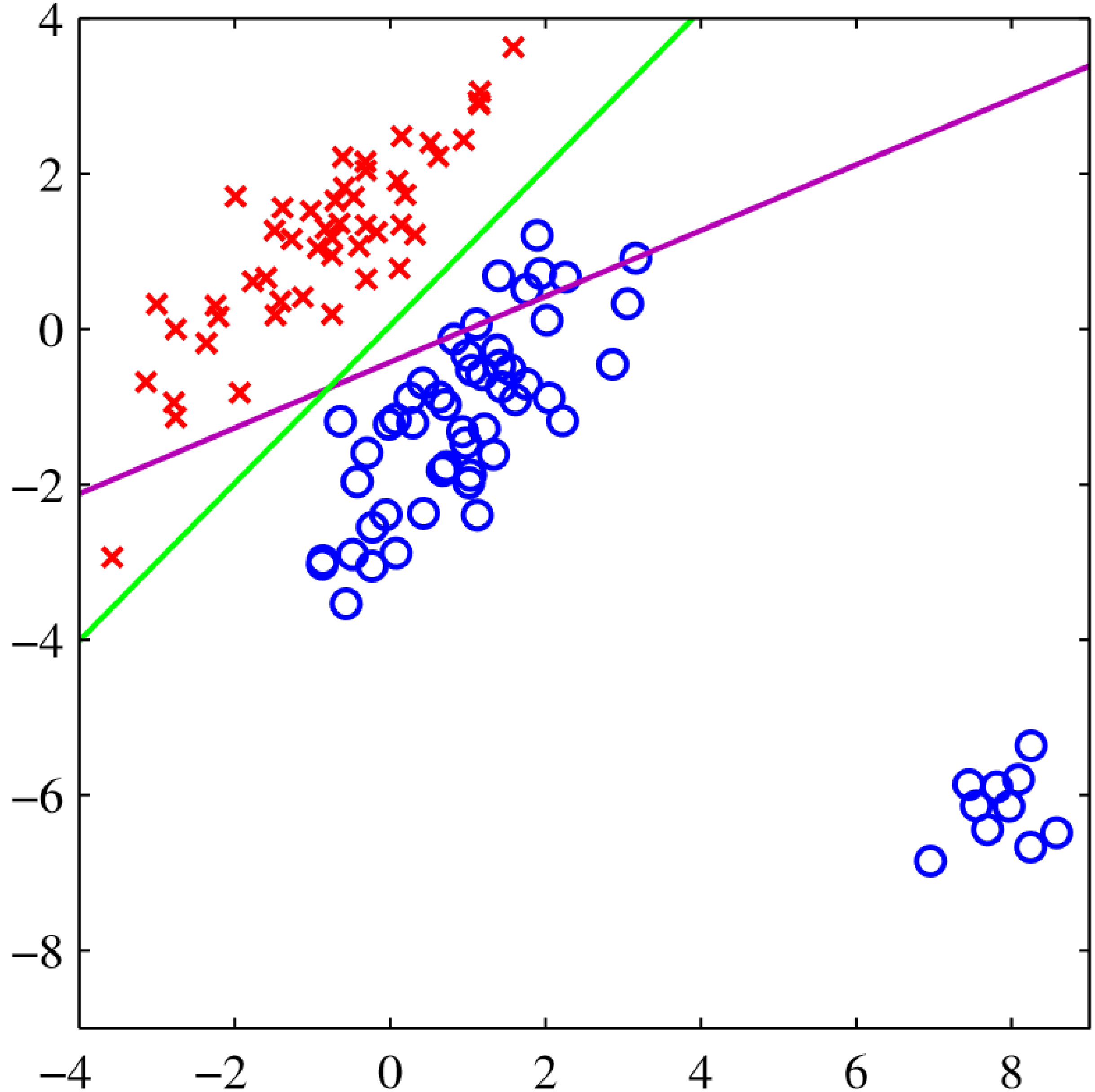
对每一个参数向量$\boldsymbol{w}_j$，可分别应用随机梯度算法迭代

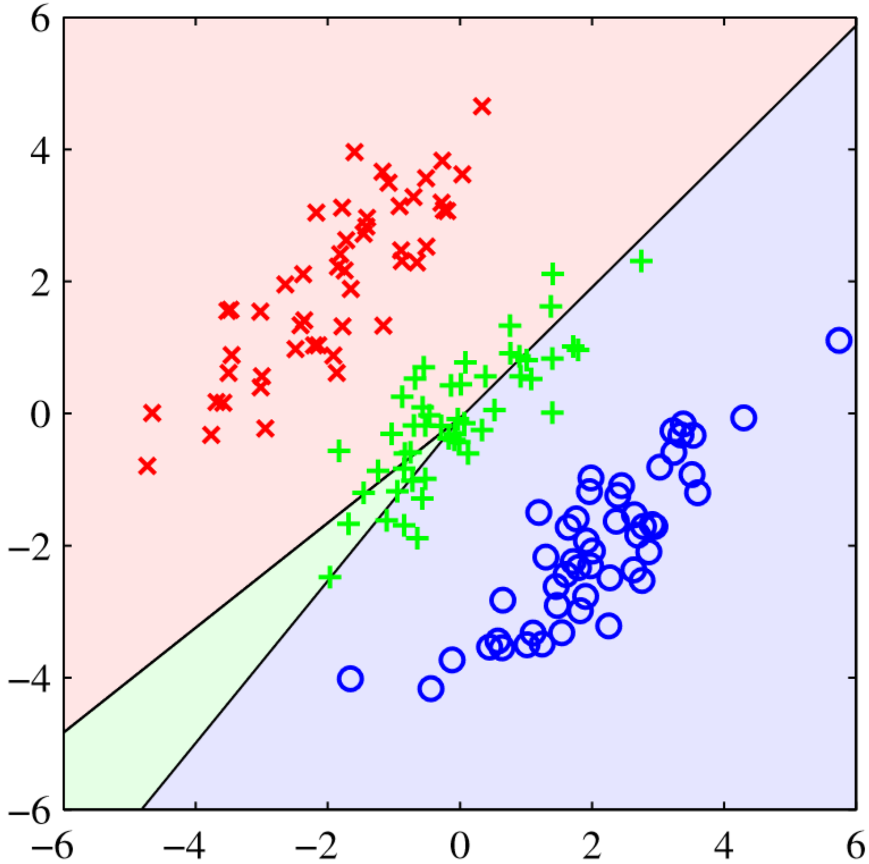
3.3 例

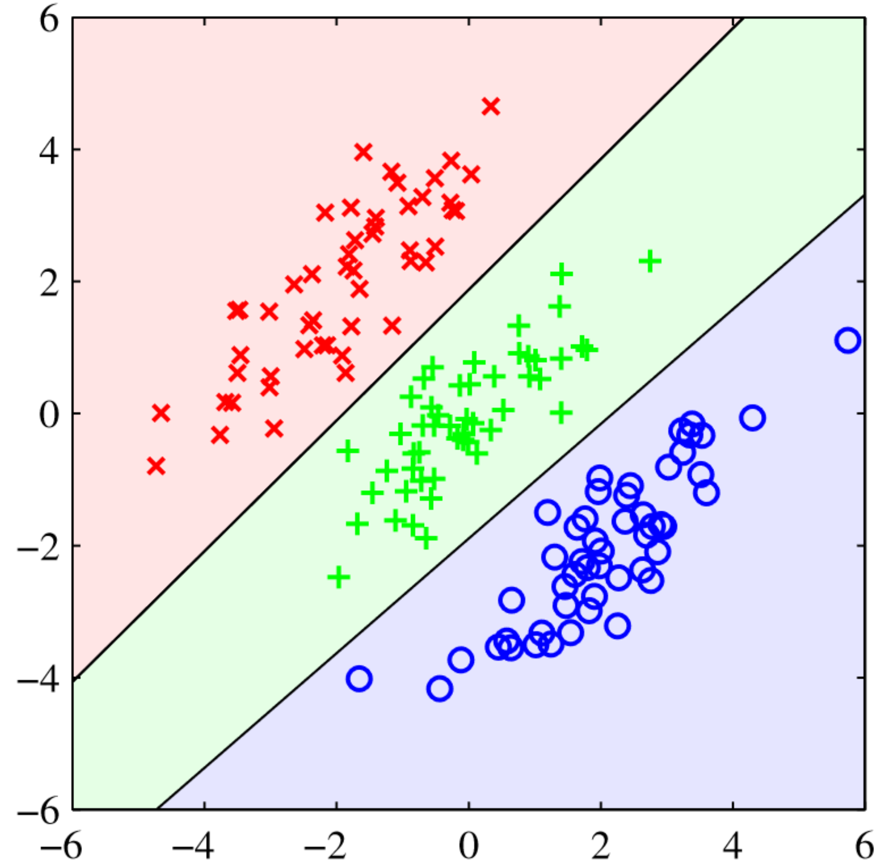
例子：两类，
分别用LS分类
和逻辑回归
绿线是逻辑回归
紫线是LS的边界



例子：右下角增加几个野点，LS分类明显变差，
逻辑回归基本不变







三类情况，即使是清晰可分的，LS分类也很差（右）
此例中，逻辑回归分类比较理想

4.分类的生成模型

若可通过数据集模型化：$p(\mathbf{x}|\mathcal{C}_k)$、$p(\mathcal{C}_k)$ 或 $p(x,\mathrm{t})$

类C1的后验概率可写为

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)+p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1+\exp(-a)} = \sigma(a) \end{aligned} \qquad \#（1）$$

其中 $$a = \ln\frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \qquad \#（2）$$

4.1离散生成模型：Naïve Bayes

输入向量的每个分量取值离散，

先假设取值（0，1），t 对应二类分类

$$\boldsymbol{x} = [x_1, x_2, \cdots, x_\mathrm{D}]^T \qquad x_\mathrm{i} \in \{0,1\}$$

例 $$\boldsymbol{x} = [0,1,1,0,1,0,0,1,0,0]^T \qquad D = 10$$

假设 $$p(\boldsymbol{x}|t) = p(x_1, x_2, \cdots, x_\mathrm{D}|t)$$

$$= \prod_{i=1}^{D} p(x_\mathrm{i}|t)$$

离散生成模型：Naïve Bayes

用t=1表示C1，t=0表示C2

用如下符号

$$\mu_{i|t=1} = \mu_{i|1} = p(x_i = 1|t = 1) = p(x_i = 1|C_1)$$

$$\mu_{i|t=0} = \mu_{i|0} = p(x_i = 1|t = 0) = p(x_i = 1|C_2)$$

类条件概率

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{D} \mu_{i|k}^{x_i} \left(1 - \mu_{i|k}\right)^{1-x_i}$$

4.2 Naïve Bayes学习

样本集 $\left\{ \mathbf{x}^{(n)}, t^{(n)} \right\}_{n=1}^{N}$

注意，区别输入向量下标，用上标表示样本序号

$$p(t = 1) = p(C_1) = \pi$$

$$p(t = 0) = p(C_2) = 1 - \pi$$

Naïve Bayes学习

联合概率表示1

$$p\left(\boldsymbol{x}^{(\mathrm{n})}, C_1\right) = p\left(\boldsymbol{x}^{(\mathrm{n})}, t = 1\right)$$

$$= p\left(C_1\right) p\left(\boldsymbol{x}^{(\mathrm{n})} \middle| C_1\right)$$

$$= \pi \prod_{i=1}^{D} \mu_{\mathrm{i}|1}^{x_{\mathrm{i}}^{(\mathrm{n})}} \left(1 - \mu_{\mathrm{i}|1}\right)^{1 - x_{\mathrm{i}}^{(\mathrm{n})}}$$

Naïve Bayes学习

联合概率表示2

$$p\left(\boldsymbol{x}^{(\mathrm{n})}, C_2\right) = p\left(\boldsymbol{x}^{(\mathrm{n})}, t = 0\right)$$

$$= p\left(C_2\right) p\left(\boldsymbol{x}^{(\mathrm{n})} \middle| C_2\right)$$

$$= \left(1 - \pi\right) \prod_{i=1}^{D} \mu_{\mathrm{i}|0}^{x_{\mathrm{i}}^{(\mathrm{n})}} \left(1 - \mu_{\mathrm{i}|0}\right)^{1 - x_{\mathrm{i}}^{(\mathrm{n})}}$$

Naïve Bayes学习

似然函数

$$p\left(\boldsymbol{t}, \boldsymbol{X} \middle| \pi, \mu_{i|1}, \mu_{i|0}\right)$$

$$= \prod_{n=1}^{N} \left(\pi \prod_{i=1}^{D} \mu_{i|1}^{x_i^{(n)}} \left(1 - \mu_{i|1}\right)^{1-x_i^{(n)}} \right)^{t^{(n)}} \times$$

$$\times \left(\left(1 - \pi\right) \prod_{i=1}^{D} \mu_{i|0}^{x_i^{(n)}} \left(1 - \mu_{i|0}\right)^{1-x_i^{(n)}} \right)^{1-t^{(n)}}$$

Naïve Bayes学习

似然函数

$$\frac{\partial \ln p\left(\boldsymbol{t}, \boldsymbol{X} \middle| \pi, \mu_{i|1}, \mu_{i|0}\right)}{\partial \pi} = 0$$

得

$$\pi = \frac{1}{N} \sum_{n=1}^{N} t^{(n)}$$

Naïve Bayes学习

似然函数

$$\frac{\partial \ln p\left(\boldsymbol{t}, \boldsymbol{X} \middle| \pi, \mu_{i|1}, \mu_{i|0}\right)}{\partial \mu_{i|1}} = 0 \qquad 得$$

$$\mu_{i|1} = \frac{\sum_{n=1}^{N} t^{(n)} x_i^{(n)}}{\sum_{n=1}^{N} t^{(n)}}$$

Naïve Bayes学习

似然函数

$$\frac{\partial \ln p\left(\boldsymbol{t}, \boldsymbol{X} \middle| \pi, \mu_{i|1}, \mu_{i|0}\right)}{\partial \mu_{i|0}} = 0 \qquad \text{得}$$

$$\mu_{i|0} = \frac{\sum_{n=1}^{N} \left(1 - t^{(n)}\right) x_i^{(n)}}{\sum_{n=1}^{N} 1 - t^{(n)}}$$

Naïve Bayes学习

$x_i^{(n)}$ 是二元变量，故参数学习算法重写为

$$\pi = \frac{1}{N}\sum_{n=1}^{N} 1\{t^{(n)} = 1\}$$

$$\mu_{i|1} = \frac{\sum_{n=1}^{N} \mathrm{I}\{t^{(n)} = 1 \cap x_i^{(n)} = 1\}}{\sum_{n=1}^{N} \mathrm{I}\{t^{(n)} = 1\}}$$

$$\mu_{i|0} = \frac{\sum_{n=1}^{N} \mathrm{I}\{t^{(n)} = 0 \cap x_i^{(n)} = 1\}}{\sum_{n=1}^{N} \mathrm{I}\{t^{(n)} = 0\}}$$

4.3 Naïve Bayes推断

给出新输入 $\boldsymbol{x}$ ，进行分类

$$p(t=1|\boldsymbol{x}) = p(C_1|\boldsymbol{x})$$

$$= \frac{p(\boldsymbol{x}|t=1)p(t=1)}{p(\boldsymbol{x})}$$

$$= \frac{\left(\prod_{i=1}^{D} p(x_i|t=1)\right)p(t=1)}{\left(\prod_{i=1}^{D} p(x_i|t=1)\right)p(t=1) + \left(\prod_{i=1}^{D} p(x_i|t=0)\right)p(t=0)}$$

Naïve Bayes分类

可推广到输入各分量取值为M个值的情况

带入学习得到的参数，后验类概率为

$$p(t=1|\mathbf{x})$$

$$= \frac{\pi \prod_{i=1}^{D} \mu_{i|1}^{x_i} (1-\mu_{i|1})^{1-x_i}}{\pi \prod_{i=1}^{D} \mu_{i|1}^{x_i} (1-\mu_{i|1})^{1-x_i} + (1-\pi) \prod_{i=1}^{D} \mu_{i|0}^{x_i} (1-\mu_{i|0})^{1-x_i}}$$

$$p(t=0|\mathbf{x}) = 1 - p(t=1|\mathbf{x})$$

4.4 拉普拉斯平滑克服零概率比值问题

Naïve Bayes分类存在的一个问题

若存在 $\mu_{i|1} = 0$ 和 $\mu_{j|0} = 0$ 则,

$$p(t = 1|\boldsymbol{x}) = \frac{0}{0}$$

无法做出判断

希望不存在 $\mu_{i|k} = 0$

拉普拉斯平滑（Laplace smoothing）

设 $z \in \{1, \ldots, k\}$

且定义 $\phi_i = p(z = i)$

有样本集 $\{z^{(1)}, \ldots, z^{(m)}\}$

标准ML估计：

$$\phi_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}{m}$$

拉普拉斯平滑为

$$\phi_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\} + 1}{m + k}$$

Naïve Bayes参数学习改进为

拉普拉斯平滑参数估计

$$\pi = \frac{1}{N}\sum_{n=1}^{N} 1\left\{t^{(n)} = 1\right\}$$

$$\mu_{i|1} = \frac{\sum_{n=1}^{N} \mathrm{I}\left\{t^{(n)} = 1 \cap x_i^{(n)} = 1\right\} + 1}{\sum_{n=1}^{N} \mathrm{I}\left\{t^{(n)} = 1\right\} + 2}$$

$$\mu_{i|0} = \frac{\sum_{n=1}^{N} \mathrm{I}\left\{t^{(n)} = 0 \cap x_i^{(n)} = 1\right\} + 1}{\sum_{n=1}^{N} \mathrm{I}\left\{t^{(n)} = 0\right\} + 2}$$