模式识别与机器学习

分类的线性模型

张晓平

武汉大学数学与统计学院

Table of contents

引言

引言

分类

把输入变量 $\boldsymbol{x}$ 分到 K 个类别 $\mathcal{C}_k$ 中的某一类。

- 将输入空间划分为不同的**决策区域** (decision region)，其边界称为**决策边界** (decision boundary)或**决策面** (decision surface)。
- **分类线性模型**：决策面是输入变量 $\boldsymbol{x}$ 的线性函数，被定义为 D 维输入空间中的 $D-1$ 维超平面。
- 若数据集可被线性决策面精确地分类，则称该数据集是**线性可分** (linearly separable)的。

引言

目标向量 $\boldsymbol{t}$ 通常表示类别标签，有不同的表达方式。

- 对于二分类问题，通常使用二元表示法。此时，目标变量 $t \in \{0, 1\}$，其中 $t = 1$ 表示类别 $\mathcal{C}_1$，$t = 0$ 表示类别 $\mathcal{C}_2$。

  t 可看做是分类结果为 $\mathcal{C}_1$ 的概率，它取极端值 0 或 1。

引言

目标向量 $\boldsymbol{t}$ 通常表示类别标签，有不同的表达方式。

- 对于二分类问题，通常使用二元表示法。此时，目标变量 $t \in \{0, 1\}$，其中 $t = 1$ 表示类别 $\mathcal{C}_1$，$t = 0$ 表示类别 $\mathcal{C}_2$。

  t 可看做是分类结果为 $\mathcal{C}_1$ 的概率，它取极端值 0 或 1。

- 对于多分类问题 ($K > 2$)，通常使用 “1-of-K” 编码 (也称 one hot encoding)。此时，$\boldsymbol{t} \in \mathbb{R}^K$，如果其类别为 $\mathcal{C}_j$，则

$$\boldsymbol{t} = \boldsymbol{e}_j = (0, \cdots, 1, \cdots, 0)^T \in \mathbb{R}^K$$

  t_k 可看做是分类结果为 $\mathcal{C}_k$ 的概率。

引言

分类问题的三种方法

- 构造**判别函数** (discriminant function)，直接把 $\boldsymbol{x}$ 分到对应的类别中。

引言

分类问题的三种方法

- 构造**判别函数** (discriminant function)，直接把 $\boldsymbol{x}$ 分到对应的类别中。

- 在**推理阶段**对条件概率分布 $p(\mathcal{C}_k \mid \boldsymbol{x})$ 建模，然后使用它进行最优**决策**。

引言

分类问题的三种方法

- 构造**判别函数** (discriminant function)，直接把 $\boldsymbol{x}$ 分到对应的类别中。

- 在**推理阶段**对条件概率分布 $p(\mathcal{C}_k \mid \boldsymbol{x})$ 建模，然后使用它进行最优**决策**。

  $p(\mathcal{C}_k \mid \boldsymbol{x})$ 的确定有两种方式：

  - **直接对 $p(\mathcal{C}_k \mid \boldsymbol{x})$ 建模**：将 $p(\mathcal{C}_k \mid \boldsymbol{x})$ 表示为参数模型，然后使用训练集来优化参数；
  - **生成式的方法**：对类条件概率密度 $p(\boldsymbol{x} \mid \mathcal{C}_k)$ 和类的先验概率分布 $p(\mathcal{C}_k)$ 建模，然后使用贝叶斯公式计算后验概率分布

$$p(\mathcal{C}_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})} \tag{1}$$

引言

分类的线性模型可表示为

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0) \tag{2}$$

其中

- $f(\cdot)$ 被称为激活函数，通常为非线性函数。
- 决策面对应于 $y(\boldsymbol{x}) = \text{const}$，即 $\boldsymbol{w}^T\boldsymbol{x} + w_0 = \text{const}$。

注

由于引入了激活函数，模型(2)不再是参数的线性模型，这会导致其计算比线性回归模型更加复杂。

判別函数 (discriminant function)

判别函数 (discriminant function)

定义

判别函数是一个以 $\boldsymbol{x}$ 为输入，把它分到类别 $\mathcal{C}_k$ 的函数。

这里只考虑线性判别函数 (linear discriminant function)，即那些决策面为超平面的判别函数。

判别函数 (discriminant function)

二分类问题

二分类问题

线性判别函数最简单的形式为

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{3}$$

其中

- $\boldsymbol{w}$ 为权向量 (weight vector)；
- w_0 为偏置 (bias)，$-w_0$ 有时被称为阈值 (threshold)。

二分类问题

线性判别函数最简单的形式为

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{3}$$

其中

- $\boldsymbol{w}$ 为权向量 (weight vector)；
- w_0 为偏置 (bias)，$-w_0$ 有时被称为阈值 (threshold)。

对于 $\boldsymbol{x}$，若 $y(\boldsymbol{x}) \geq 0$，则它被分到 $\mathcal{C}_1$，否则被分到 $\mathcal{C}_2$。

二分类问题

线性判别函数最简单的形式为

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{3}$$

其中

- $\boldsymbol{w}$ 为权向量 (weight vector)；
- w_0 为偏置 (bias)，$-w_0$ 有时被称为阈值 (threshold)。

对于 $\boldsymbol{x}$，若 $y(\boldsymbol{x}) \geq 0$，则它被分到 $\mathcal{C}_1$，否则被分到 $\mathcal{C}_2$。

因此，对应的决策面由 $y(\boldsymbol{x}) = 0$ 确定，它对应于 D 维空间中的一个 $D-1$ 维超平面。

二分类问题

- 考虑决策面上的任意两点 $\boldsymbol{x}_A$ 和 $\boldsymbol{x}_B$，因 $y(\boldsymbol{x}_A) = y(\boldsymbol{x}_B) = 0$，故

$$\boldsymbol{w}^T(\boldsymbol{x}_A - \boldsymbol{x}_B) = 0$$

这说明 $\boldsymbol{w}$ 与决策面上的任何向量都正交，从而$\boldsymbol{w}$ 决定了决策面的方向。

二分类问题

- 考虑决策面上的任意两点 $\boldsymbol{x}_A$ 和 $\boldsymbol{x}_B$，因 $y(\boldsymbol{x}_A) = y(\boldsymbol{x}_B) = 0$，故

$$\boldsymbol{w}^T(\boldsymbol{x}_A - \boldsymbol{x}_B) = 0$$

  这说明 $\boldsymbol{w}$ 与决策面上的任何向量都正交，从而$\boldsymbol{w}$ 决定了决策面的方向。

- 若 $\boldsymbol{x}$ 是决策面 $\mathcal{S}$ 上的一个点，即 $y(\boldsymbol{x}) = 0$，则原点 $\boldsymbol{O}$ 到决策面的距离为

$$d(\boldsymbol{O}, \mathcal{S}) = \frac{w_0}{\|\boldsymbol{w}\|} = \frac{\boldsymbol{w}^T\boldsymbol{x}}{\|\boldsymbol{w}\|} \tag{4}$$

  这说明w_0 决定了决策面的位置。

二分类问题

考虑任一 $\boldsymbol{x}$，设其到决策面 $\mathcal{S}$ 上的距离为 r，投影为 $\boldsymbol{x}^{\perp}$，则有

$$\boldsymbol{x} = \boldsymbol{x}^{\perp} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{5}$$

二分类问题

考虑任一 $\boldsymbol{x}$，设其到决策面 $\mathcal{S}$ 上的距离为 r，投影为 $\boldsymbol{x}^{\perp}$，则有

$$\boldsymbol{x} = \boldsymbol{x}^{\perp} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{5}$$

两边同乘 $\boldsymbol{w}^T$，再加上 w_0，即得

$$\boldsymbol{w}^T\boldsymbol{x} + w_0 = \boldsymbol{w}^T\boldsymbol{x}^{\perp} + w_0 + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

即

$$y(\boldsymbol{x}) = y(\boldsymbol{x}^{\perp}) + r\|\boldsymbol{w}\|$$

二分类问题

考虑任一 $\boldsymbol{x}$，设其到决策面 $\mathcal{S}$ 上的距离为 r，投影为 $\boldsymbol{x}^{\perp}$，则有

$$\boldsymbol{x} = \boldsymbol{x}^{\perp} + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{5}$$

两边同乘 $\boldsymbol{w}^T$，再加上 w_0，即得

$$\boldsymbol{w}^T\boldsymbol{x} + w_0 = \boldsymbol{w}^T\boldsymbol{x}^{\perp} + w_0 + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

即

$$y(\boldsymbol{x}) = y(\boldsymbol{x}^{\perp}) + r\|\boldsymbol{w}\|$$

因 $\boldsymbol{x}^{\perp}$ 在决策面 $\mathcal{S}$ 上，故 $y(\boldsymbol{x}^{\perp}) = 0$，从而有

$$r = \frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|}.$$

二分类问题

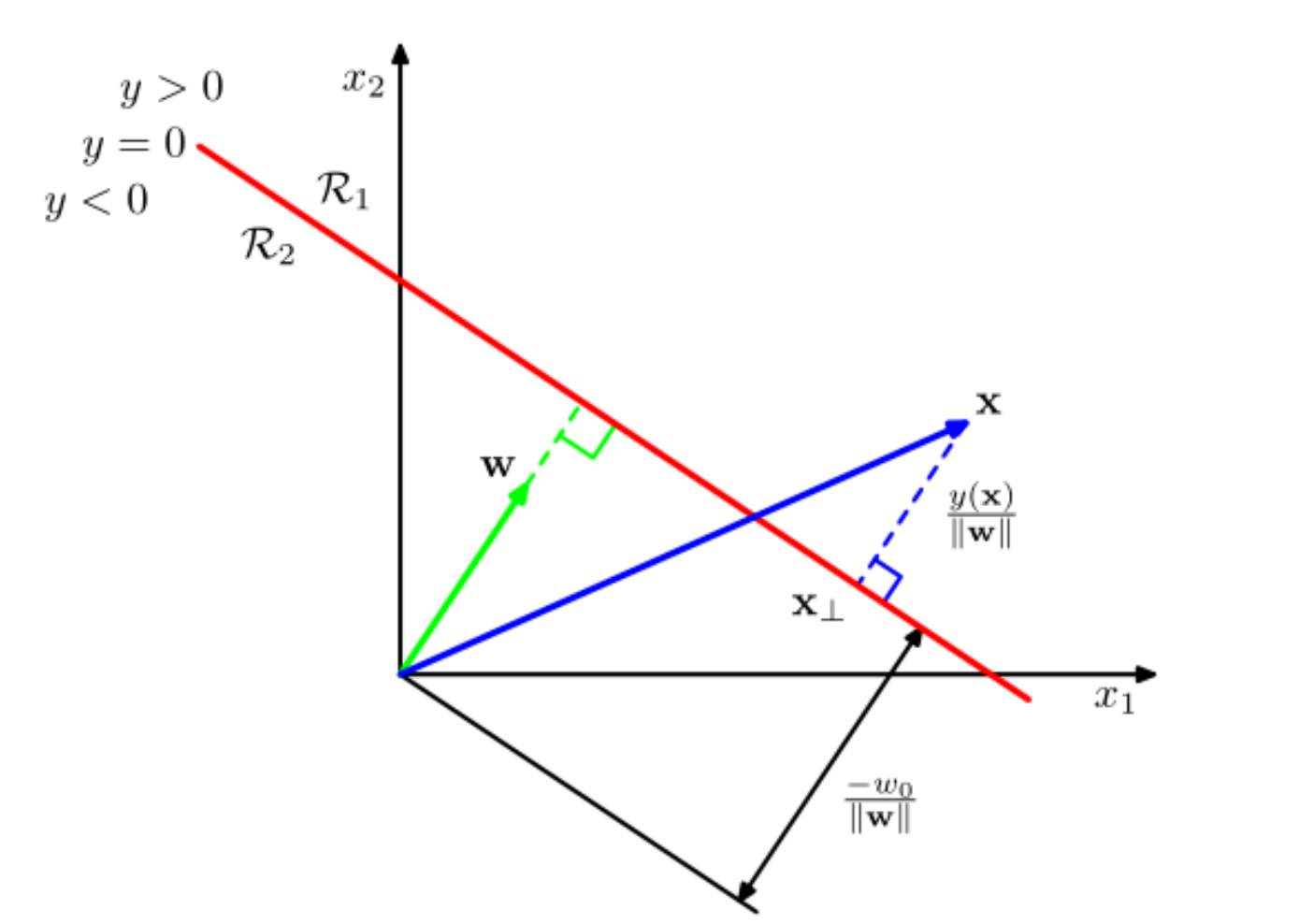


图 1: 二维线性判别函数的几何表示。决策面 (红线) 垂直于 $\boldsymbol{w}$，它距离原点的偏移量由 w_0 控制。任一点 $\boldsymbol{x}$ 到决策面的距离为 $y(\boldsymbol{x})/\|\boldsymbol{w}\|$

二分类问题

为表示方便，可定义 $\tilde{\boldsymbol{w}} = \begin{pmatrix} w_0 \\ \boldsymbol{w} \end{pmatrix}, \tilde{\boldsymbol{x}} = \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}$，则 $y(\boldsymbol{x})$ 可表示为

$$y(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}. \tag{6}$$

此时，决策面为一个 D 维超平面，它会穿过 $D+1$ 维扩展输入空间的原点。

判别函数 (discriminant function)

多分类问题 ($K > 2$)

多分类问题 ($K > 2$)

K 类判别函数可定义为

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K \tag{7}$$

多分类问题 ($K > 2$)

K 类判别函数可定义为

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K \tag{7}$$

对于 $\boldsymbol{x}$，若

$$y_k(\boldsymbol{x}) > y_j(\boldsymbol{x}), \quad j \neq k,$$

则将 $\boldsymbol{x}$ 分到类别 $\mathcal{C}_k$。

多分类问题 ($K > 2$)

K 类判别函数可定义为

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K \tag{7}$$

对于 $\boldsymbol{x}$，若

$$y_k(\boldsymbol{x}) > y_j(\boldsymbol{x}), \quad j \neq k,$$

则将 $\boldsymbol{x}$ 分到类别 $\mathcal{C}_k$。

于是，类别 $\mathcal{C}_k$ 和 $\mathcal{C}_j$ 之间的决策面为 $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$，对应于一个 $D-1$ 维超平面，其形式为

$$(\boldsymbol{w}_k - \boldsymbol{w}_j)^T \boldsymbol{x} + (w_{k0} - w_{j0}) = 0. \tag{8}$$

多分类问题 ($K > 2$)

定理

判别函数 $y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$，$k = 1, \cdots, K$ 的决策区域是单联通的，并且是凸的。

多分类问题 ($K > 2$)

定理

判别函数 $y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$，$k = 1, \cdots, K$ 的决策区域是单联通的，并且是凸的。

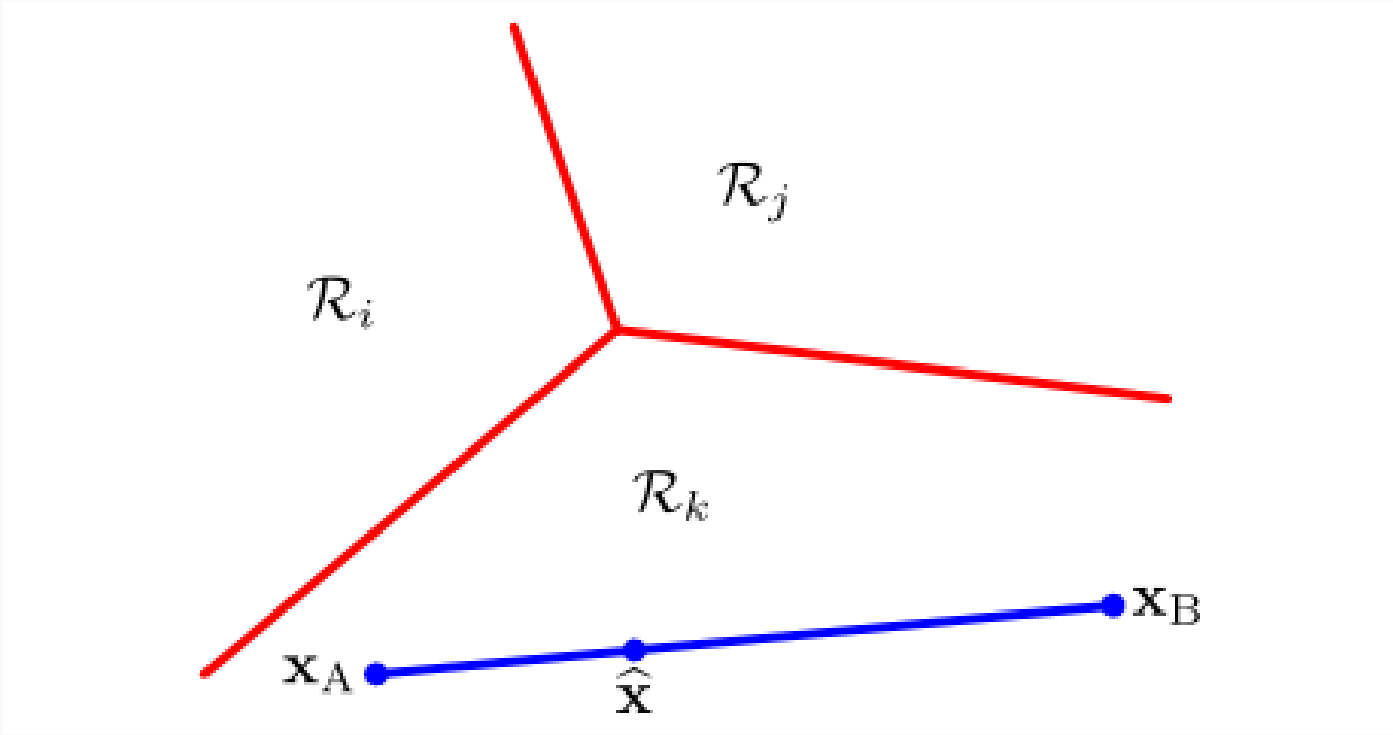


图 2: 多类判别函数的决策区域。若两点 $\boldsymbol{x}_A$ 和 $\boldsymbol{x}_B$ 在同一决策区域 $\mathcal{R}_k$ 中，则任一位于连接两点的线段上的 $\hat{\boldsymbol{x}}$ 一定在 $\mathcal{R}_k$ 内，故决策区域一定是单联通的、凸的。

多分类问题 ($K > 2$)

证明

考虑决策区域 $\mathcal{R}_k$ 中的任意两点 $\boldsymbol{x}_A, \boldsymbol{x}_B$，及

$$\hat{\boldsymbol{x}} = \lambda \boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B, \quad 0 \leq \lambda \leq 1.$$

由 $y_k(\boldsymbol{x})$ 的性质可知

$$y_k(\hat{\boldsymbol{x}}) = \lambda y_k(\boldsymbol{x}_A) + (1-\lambda) y_k(\boldsymbol{x}_B). \tag{9}$$

由于 $\boldsymbol{x}_A, \boldsymbol{x}_B \in \mathcal{R}_k$，故 $\forall j \neq k$，有 $y_k(\boldsymbol{x}_A) > y_j(\boldsymbol{x}_A)$ 及 $y_k(\boldsymbol{x}_B) > y_j(\boldsymbol{x}_B)$，因此

$$y_k(\hat{\boldsymbol{x}}) > y_j(\hat{\boldsymbol{x}}),$$

从而 $\hat{\boldsymbol{x}}$ 也在 $\mathcal{R}_k$ 内部，即 $\mathcal{R}_k$ 单联通并且为凸。

判别函数 (discriminant function)

以下将介绍三种线性判别函数的参数估计方法，即

- 最小二乘法
- Fisher 线性判别函数
- 感知机算法

判別函数 (discriminant function)

最小二乗法

最小二乘法

设每个类 $\mathcal{C}_k$ 都由自己的线性模型描述，即

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K. \tag{10}$$

最小二乘法

设每个类 $\mathcal{C}_k$ 都由自己的线性模型描述，即

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}, \quad k = 1, \cdots, K. \tag{10}$$

记

$$\tilde{\boldsymbol{w}}_k = \begin{pmatrix} w_{k0} \\ \boldsymbol{w}_k \end{pmatrix} \in \mathbb{R}^{D+1}, \quad \tilde{\boldsymbol{x}} = \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \in \mathbb{R}^{D+1}, \quad \boldsymbol{y}(\boldsymbol{x}) = \begin{pmatrix} y_1(\boldsymbol{x}) \\ \vdots \\ y_K(\boldsymbol{x}) \end{pmatrix} \in \mathbb{R}^K$$

及

$$\tilde{\boldsymbol{W}} = \begin{pmatrix} \tilde{\boldsymbol{w}}_1 & \cdots & \tilde{\boldsymbol{w}}_K \end{pmatrix} = \begin{pmatrix} \boldsymbol{w}_0^T \\ \boldsymbol{W} \end{pmatrix} \in \mathbb{R}^{(D+1)\times K},$$

$$\boldsymbol{w}_0 = \begin{pmatrix} w_{10} & \cdots & w_{K0} \end{pmatrix}^T, \quad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_K \end{pmatrix}$$

则(10)可表示为

$$\boldsymbol{y}(\boldsymbol{x}) = \tilde{\boldsymbol{W}}^T \tilde{\boldsymbol{x}} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{w}_0. \tag{11}$$

最小二乘法

设有一个训练数据集 $\{(\boldsymbol{x}_n, \boldsymbol{t}_n)\}_{n=1}^N$，定义矩阵

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{pmatrix} \in \mathbb{R}^{N\times K}, \quad \tilde{\boldsymbol{X}} = \begin{pmatrix} \tilde{\boldsymbol{x}}_1^T \\ \vdots \\ \tilde{\boldsymbol{x}}_N^T \end{pmatrix} \in \mathbb{R}^{N\times (D+1)}$$

则平方和误差函数可表示为

$$\begin{aligned} E_{\mathcal{D}}(\tilde{\boldsymbol{W}}) &= \sum_{n=1}^{N} \|\boldsymbol{y}(\boldsymbol{x}_n) - \boldsymbol{t}_n\|^2 \\ &= \sum_{n=1}^{N} \left(\tilde{\boldsymbol{W}}^T\tilde{\boldsymbol{x}} - \boldsymbol{t}_n\right)\left(\tilde{\boldsymbol{W}}^T\tilde{\boldsymbol{x}} - \boldsymbol{t}_n\right)^T \\ &= \frac{1}{2}\mathrm{tr}\left\{(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})^T(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})\right\} \end{aligned} \tag{12}$$

最小二乘法

它关于 $\tilde{\boldsymbol{W}}$ 的导数为

$$\frac{\partial E_{\mathcal{D}}(\tilde{\boldsymbol{W}})}{\partial \tilde{\boldsymbol{W}}} = \tilde{\boldsymbol{X}}^T(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})$$

置其为零可得

$$\tilde{\boldsymbol{W}} = (\boldsymbol{X}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}\boldsymbol{T} := \tilde{\boldsymbol{X}}^{\dagger}\boldsymbol{T}$$

其中 $\tilde{\boldsymbol{X}}^{\dagger}$ 为 $\tilde{\boldsymbol{X}}$ 的伪逆。

最小二乘法

它关于 $\tilde{\boldsymbol{W}}$ 的导数为

$$\frac{\partial E_{\mathcal{D}}(\tilde{\boldsymbol{W}})}{\partial \tilde{\boldsymbol{W}}} = \tilde{\boldsymbol{X}}^T(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}} - \boldsymbol{T})$$

置其为零可得

$$\tilde{\boldsymbol{W}} = (\boldsymbol{X}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}\boldsymbol{T} := \tilde{\boldsymbol{X}}^{\dagger}\boldsymbol{T}$$

其中 $\tilde{\boldsymbol{X}}^{\dagger}$ 为 $\tilde{\boldsymbol{X}}$ 的伪逆。

于是判别函数(10)的形式为

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{T}^T\left(\tilde{\boldsymbol{X}}^{\dagger}\right)^T\tilde{\boldsymbol{x}}. \tag{13}$$

最小二乘法

性质

对于最小二乘法，若训练集中的每个目标向量 $\boldsymbol{t}_n$ 满足线性约束

$$\boldsymbol{a}^T \boldsymbol{t}_n + b = 0, \quad n = 1, \cdots, N, \tag{14}$$

其中 $\boldsymbol{a} \in \mathbb{R}^K$ 和 b 为常数，则 $\forall \boldsymbol{x}$，模型的预测也满足同样的约束，即

$$\boldsymbol{a}^T \boldsymbol{y}(\boldsymbol{x}) + b = 0. \tag{15}$$

最小二乘法

证明

$E_{\mathcal{D}}(\tilde{\boldsymbol{W}})$ 可改写为

$$E_{\mathcal{D}}(\boldsymbol{w}_0, \boldsymbol{W}) = \frac{1}{2}\mathrm{tr}\left\{(\boldsymbol{X}\boldsymbol{W} + \mathbf{1}\boldsymbol{w}_0^T - \boldsymbol{T})^T(\boldsymbol{X}\boldsymbol{W} + \mathbf{1}\boldsymbol{w}_0^T - \boldsymbol{T})\right\}, \quad (16)$$

其中

$$\mathbf{1} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^T \in \mathbb{R}^N.$$

对其关于 $\boldsymbol{w}_0$ 求导可得

$$2N\boldsymbol{w}_0 + 2(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{T})^T\mathbf{1}$$

置其为零可求得

$$\boldsymbol{w}_0 = \bar{\boldsymbol{t}} - \boldsymbol{W}^T\bar{\boldsymbol{x}}$$

其中

$$\bar{\boldsymbol{t}} = \frac{1}{N}\boldsymbol{T}^T\mathbf{1}, \quad \bar{\boldsymbol{x}} = \frac{1}{N}\boldsymbol{X}^T\mathbf{1}$$

最小二乘法

证明

将其代入(16)可得

$$E_{\mathcal{D}}(\boldsymbol{W}) = \frac{1}{2}\mathrm{tr}\left\{(\boldsymbol{X}\boldsymbol{W} + \bar{\boldsymbol{T}} - \bar{\boldsymbol{X}}\boldsymbol{W} - \boldsymbol{T})^T(\boldsymbol{X}\boldsymbol{W} + \bar{\boldsymbol{T}} - \bar{\boldsymbol{X}}\boldsymbol{W} - \boldsymbol{T})\right\},$$

其中

$$\bar{\boldsymbol{T}} = \mathbf{1}\bar{\boldsymbol{t}}^T, \quad \bar{\boldsymbol{X}} = \mathbf{1}\bar{\boldsymbol{x}}^T.$$

对 $E_{\mathcal{D}}(\boldsymbol{W})$ 关于 $\boldsymbol{W}$ 求导并置其为零可得

$$\boldsymbol{W} = \left(\hat{\boldsymbol{X}}^T\hat{\boldsymbol{X}}\right)^{-1}\hat{\boldsymbol{X}}^T\hat{\boldsymbol{T}} = \hat{\boldsymbol{X}}^\dagger\hat{\boldsymbol{T}}$$

其中

$$\hat{\boldsymbol{X}} = \boldsymbol{X} - \bar{\boldsymbol{X}}, \quad \hat{\boldsymbol{T}} = \boldsymbol{T} - \bar{\boldsymbol{T}}.$$

最小二乘法

证明

考虑一个新的输入 $\boldsymbol{x}^*$ 的预测，

$$\begin{aligned} \boldsymbol{y}(\boldsymbol{x}^*) &= \boldsymbol{W}^T\boldsymbol{x}^* + \boldsymbol{w}_0 \\ &= \boldsymbol{W}^T\boldsymbol{x}^* + \bar{\boldsymbol{t}} - \boldsymbol{W}^T\bar{\boldsymbol{x}} \\ &= \bar{\boldsymbol{t}} + \hat{\boldsymbol{T}}^T\left(\hat{\boldsymbol{X}}^\dagger\right)^T(\boldsymbol{x}^* - \bar{\boldsymbol{x}}) \end{aligned}$$

而(14)可改写为

$$\boldsymbol{a}^T\boldsymbol{T}^T + b\mathbf{1}^T = 0,$$

两端同时右乘 $\mathbf{1}$ 可得

$$\boldsymbol{a}^T\boldsymbol{T}^T\mathbf{1} + Nb = 0,$$

于是

$$\boldsymbol{a}^T\bar{\boldsymbol{t}} = \frac{1}{N}\boldsymbol{a}^T\boldsymbol{T}^T\mathbf{1} = -b.$$

证明

因此

$$\begin{aligned} \boldsymbol{a}^T \boldsymbol{y}(\boldsymbol{x}^*) &= \boldsymbol{a}^T \bar{\boldsymbol{t}} + \boldsymbol{a}^T \hat{\boldsymbol{T}}^T \left(\hat{\boldsymbol{X}}^\dagger\right)^T (\boldsymbol{x}^* - \bar{\boldsymbol{x}}) \\ &= \boldsymbol{a}^T \boldsymbol{t} = -b, \end{aligned}$$

这里用到了 $\boldsymbol{a}^T \hat{\boldsymbol{T}}^T = \boldsymbol{a}^T (\boldsymbol{T} - \bar{\boldsymbol{T}})^T = b(\mathbf{1} - \mathbf{1})^T = \mathbf{0}^T$.

最小二乘法

推论

对于 K 分类问题，若 $\boldsymbol{t}$ 使用的是 “one hot encoding”，则 $\forall \boldsymbol{x}$ 有

$$\sum_{i=1}^{K} y_i(\boldsymbol{x}) = 1.$$

推论

对于 K 分类问题，若 $\boldsymbol{t}$ 使用的是 “one hot encoding”，则 $\forall \boldsymbol{x}$ 有

$$\sum_{i=1}^{K} y_i(\boldsymbol{x}) = 1.$$

证明.
将(14)中的 $\boldsymbol{a}$ 和 b 取为

$$\boldsymbol{a} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad b = -1$$

即可。 □

最小二乘法

最小二乘法的缺点：最小二乘法给出了判别函数参数的解析解，但是它对离群点 (outliers) 缺少鲁棒性。

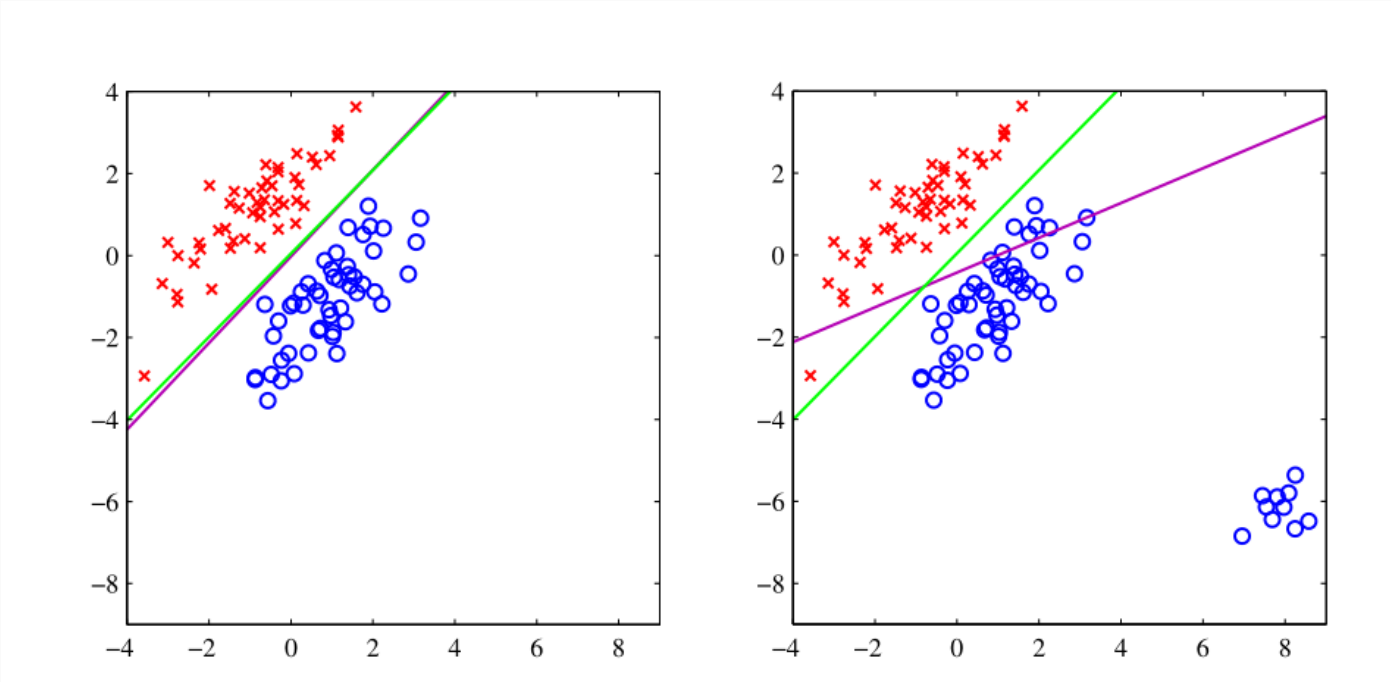


图 3: 最小二乘法用于二分类：相比与左图，右图中多了一些额外的数据，它们使得决策边界的位置发生了很大的改变，即使左图中的原始决策边界也能正确地分类。平方和误差函数惩罚了“过于正确”的预测，因为它们在正确的一侧，并且距离决策边界太远。

最小二乘法

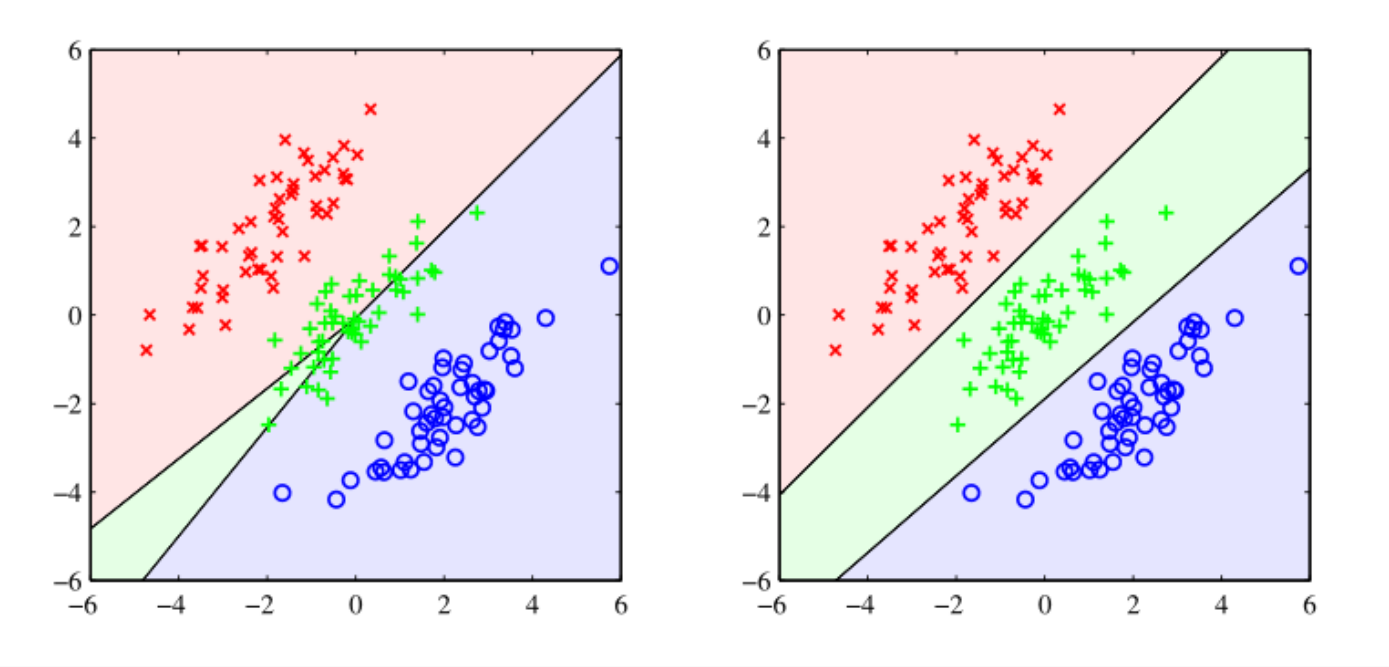


图 4: 最小二乘法用于三分类问题。左图为最小二乘方法的结果，分配到绿色类别的区域过小，大部分来自该类别的点都被错误分类；右图为 softmax 回归的结果，给出了正确分类。

最小二乘法

最小二乘法的失败并不意外，因为它是对应于高斯条件分布假设下的极大似然法，而二值目标向量的概率分布显然不是高斯分布。若使用恰当的概率模型，将会得到更好的分类方法。

判别函数 (discriminant function)

Fisher 线性判别函数

Fisher 线性判别函数

Fisher 线性判别分析，又称线性判别分析 (Linear Discriminant Analysis, LDA)，是一种监督学习的降维技术。

二分类的 Fisher 线性判别函数

设有输入向量 $\boldsymbol{x} \in \mathbb{R}^D$，使用下式投影到一维空间：

$$y = \boldsymbol{w}^T \boldsymbol{x}. \tag{17}$$

二分类的 Fisher 线性判别函数

设有输入向量 $\boldsymbol{x} \in \mathbb{R}^D$，使用下式投影到一维空间：

$$y = \boldsymbol{w}^T \boldsymbol{x}. \tag{17}$$

对 y 设定一个阈值 $-w_0$，

- 若 $y \geq -w_0$，则把样本分到 $\mathcal{C}_1$；
- 若 $y < -w_0$，则把样本分到 $\mathcal{C}_2$。

此即标准的线性分类器。

二分类的 Fisher 线性判别函数

设有输入向量 $\boldsymbol{x} \in \mathbb{R}^D$，使用下式投影到一维空间：

$$y = \boldsymbol{w}^T \boldsymbol{x}. \tag{17}$$

对 y 设定一个阈值 $-w_0$，

- 若 $y \geq -w_0$，则把样本分到 $\mathcal{C}_1$；
- 若 $y < -w_0$，则把样本分到 $\mathcal{C}_2$。

此即标准的线性分类器。

注

一般来说，将高维样本投影到一维空间会造成信息的丢失，从而导致在高维空间中可分的样本在一维空间中不可分。但是，通过调整 $\boldsymbol{w}$，可选择一个让类别之间间隔最大的投影。

二分类的 Fisher 线性判别函数

考虑二分类问题，设属于 $\mathcal{C}_1$ 的样本个数为 N_1，属于 $\mathcal{C}_2$ 的样本个数为 N_2，则两类的均值向量分别为

$$\boldsymbol{m}_1 = \frac{1}{N_1} \sum_{i \in \mathcal{C}_1} \boldsymbol{x}_i, \quad \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{i \in \mathcal{C}_2} \boldsymbol{x}_i. \tag{18}$$

二分类的 Fisher 线性判别函数

考虑二分类问题，设属于 C_1 的样本个数为 N_1，属于 C_2 的样本个数为 N_2，则两类的均值向量分别为

$$\boldsymbol{m}_1 = \frac{1}{N_1}\sum_{i\in C_1}\boldsymbol{x}_i, \quad \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{i\in C_2}\boldsymbol{x}_i. \tag{18}$$

最简单的度量类别之间间隔最大的方式是计算类别均值投影后的距离，

即求 $\boldsymbol{w}^*$ 使得

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} \ \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) \quad \text{s.t.} \quad \|\boldsymbol{w}\|_2^2 = 1$$

二分类的 Fisher 线性判别函数

该优化问题可利用 Lagrange 乘子法求解，具体步骤如下：

- 构造 Lagrange 函数

$$L(\boldsymbol{w};\lambda) = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) + \lambda(\|\boldsymbol{w}\|_2^2 - 1)$$

- 计算梯度

$$\begin{aligned} \frac{L(\boldsymbol{w};\lambda)}{\partial \boldsymbol{w}} &= \boldsymbol{m}_2 - \boldsymbol{m}_1 + 2\lambda \boldsymbol{w} \\ \frac{L(\boldsymbol{w};\lambda)}{\partial \lambda} &= \|\boldsymbol{w}\|_2^2 - 1 \end{aligned}$$

- 置其为零可得

$$\boldsymbol{w}^* \propto \boldsymbol{m}_2 - \boldsymbol{m}_1.$$

二分类的 Fisher 线性判别函数

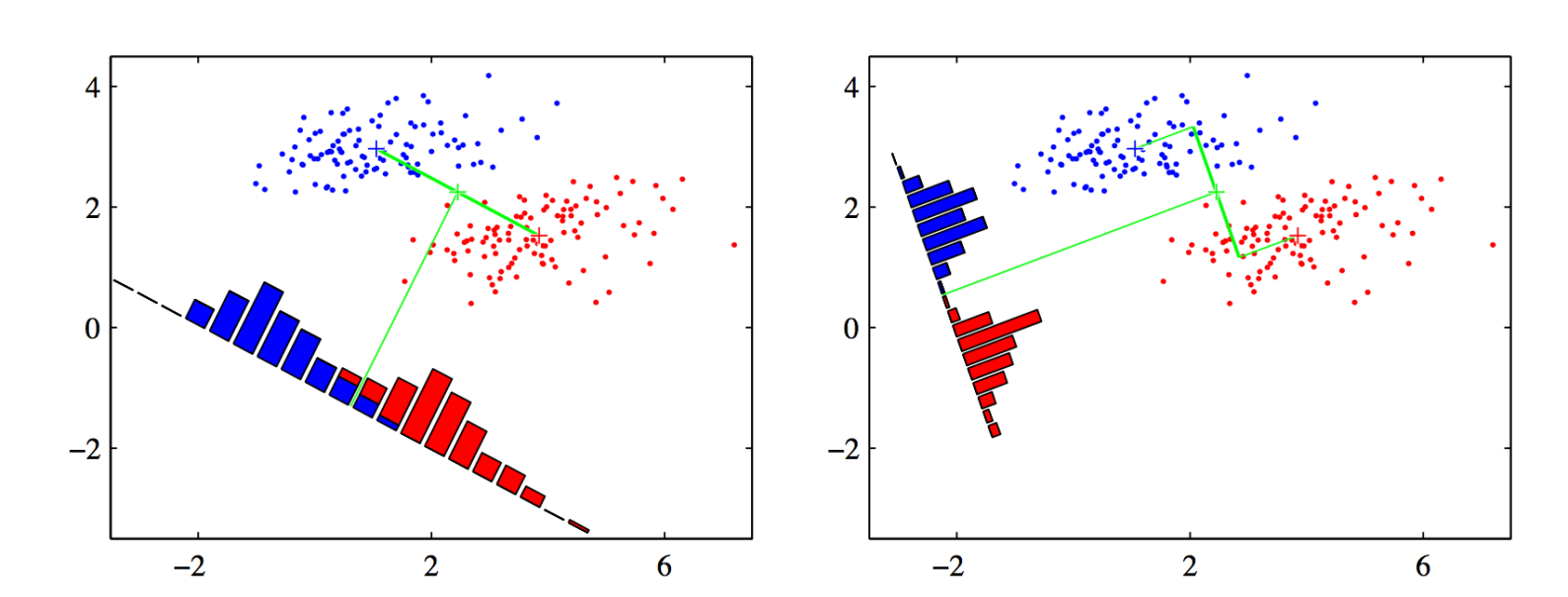
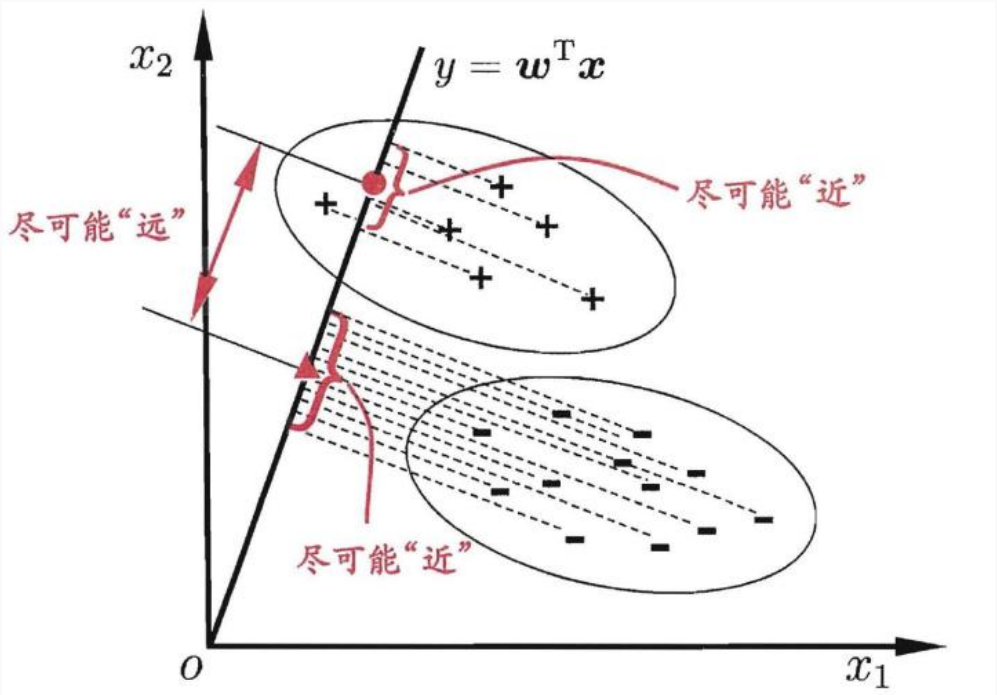


图 5: 图中的样本在二维空间中可被完美的分开。左图中，所有样本点投影到连接两类别均值的直线上，出现了一定程度的重叠；右图的投影是基于 Fisher 线性判别准则得到的，在分类效果上有极大的提升。

二分类的 Fisher 线性判别函数

Fisher 线性判别的思想

给定训练数据集，设法将样本投影到一条直线上，使得同类样本的投影点尽可能接近、异类样本的投影点尽可能地远离；对新样本分类时，将其投影到这条直线上，再根据投影点的位置来确定新样本的类别。



二分类的 Fisher 线性判别函数

投影公式

$$y = \boldsymbol{w}^T \boldsymbol{x}$$

将 $\boldsymbol{x} \in \mathbb{R}^D$ 投影到一维空间 $y \in \mathbb{R}$。来自类别 $\mathcal{C}_k$ 的样本经投影后的类内方差为

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2,$$

其中

$$\boldsymbol{y}_n = \boldsymbol{w}^T \boldsymbol{x}_n, \quad m_k = \boldsymbol{w}^T \boldsymbol{m}_k,$$

于是，可定义总的类内方差为

$$s_1^2 + s_2^2.$$

二分类的 Fisher 线性判别函数

投影公式

$$y = \boldsymbol{w}^T \boldsymbol{x}$$

将 $\boldsymbol{x} \in \mathbb{R}^D$ 投影到一维空间 $y \in \mathbb{R}$。来自类别 $\mathcal{C}_k$ 的样本经投影后的类内方差为

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2,$$

其中

$$\boldsymbol{y}_n = \boldsymbol{w}^T \boldsymbol{x}_n, \quad m_k = \boldsymbol{w}^T \boldsymbol{m}_k,$$

于是，可定义总的类内方差为

$$s_1^2 + s_2^2.$$

Fisher 准则根据类间方差和类内方差的比值来定义，即

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{19}$$

二分类的 Fisher 线性判别函数

它也可显式地写成

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}} \tag{20}$$

其中

$$\begin{aligned} \boldsymbol{S}_B &= (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \\ \boldsymbol{S}_W &= \sum_{n \in \mathcal{C}_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T \end{aligned} \tag{21}$$

分别表示类间 (between-class) 协方差矩阵和类内 (within-class) 协方差矩阵。

二分类的 Fisher 线性判别函数

对 $J(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 求导可得

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 2\frac{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2},$$

二分类的 Fisher 线性判别函数

对 $J(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 求导可得

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 2\frac{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2},$$

置其为零可得 $J(\boldsymbol{w})$ 取得极大值的条件为

$$(\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w} = (\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} \tag{22}$$

二分类的 Fisher 线性判别函数

对 $J(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 求导可得

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 2\frac{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2},$$

置其为零可得 $J(\boldsymbol{w})$ 取得极大值的条件为

$$(\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w} = (\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} \tag{22}$$

- 一方面，由 $\boldsymbol{S}_B$ 的定义可知

$$\boldsymbol{S}_B\boldsymbol{w} \propto \boldsymbol{m}_2 - \boldsymbol{m}_1,$$

- 另一方面，我们只关注 $\boldsymbol{w}$ 的方向，可以忽略(22)中的标量因子，于是

$$\boldsymbol{S}_W\boldsymbol{w} \propto \boldsymbol{S}_B\boldsymbol{w}$$

二分类的 Fisher 线性判别函数

对 $J(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 求导可得

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 2\frac{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} - (\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w}}{(\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})^2},$$

置其为零可得 $J(\boldsymbol{w})$ 取得极大值的条件为

$$(\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w})\boldsymbol{S}_W\boldsymbol{w} = (\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w})\boldsymbol{S}_B\boldsymbol{w} \tag{22}$$

- 一方面，由 $\boldsymbol{S}_B$ 的定义可知

$$\boldsymbol{S}_B\boldsymbol{w} \propto \boldsymbol{m}_2 - \boldsymbol{m}_1,$$

- 另一方面，我们只关注 $\boldsymbol{w}$ 的方向，可以忽略(22)中的标量因子，于是

$$\boldsymbol{S}_W\boldsymbol{w} \propto \boldsymbol{S}_B\boldsymbol{w}$$

综上可知，

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1), \tag{23}$$

它被称为Fisher 线性判别函数 (Fisher Linear Discriminant)。

二分类的 Fisher 线性判别函数

注

若类内协方差矩阵各项同性，即 $\boldsymbol{S}_W \propto \boldsymbol{I}$，则 $\boldsymbol{w} \propto \boldsymbol{m}_2 - \boldsymbol{m}_1$，即 $\boldsymbol{w}$ 正比于类均值的差。

二分类的 Fisher 线性判别函数

注

若类内协方差矩阵各项同性，即 $\boldsymbol{S}_W \propto \boldsymbol{I}$，则 $\boldsymbol{w} \propto \boldsymbol{m}_2 - \boldsymbol{m}_1$，即 $\boldsymbol{w}$ 正比于类均值的差。

注

严格来说，(23)并不是一个判别函数，而是对数据向一维投影方向的一个具体选择。有了投影方向，就可以构造判别函数，构造方式如下：

- 选择一个阈值 y_0，当 $y(\boldsymbol{x}) \geq y_0$ 时，把数据 $\boldsymbol{x}$ 分到 $\mathcal{C}_1$；
- 否则，把数据 $\boldsymbol{x}$ 分到 $\mathcal{C}_2$。

与最小二乘法的关系

对于分类问题，

- 最小二乘法的目标是使得模型的预测尽可能地与目标值接；
- Fisher 判别准则的目标时使得输出空间的类别有最大的区分度。

以下将说明对于二分类问题，Fisher 判别准则是最小二乘法的一个特例。

与最小二乘法的关系

我们修改一下目标值的表示方法：

$$t_n = \begin{cases} \frac{N}{N_1}, & n \in \mathcal{C}_1 \\ -\frac{N}{N_2}, & n \in \mathcal{C}_2 \end{cases} \tag{24}$$

其中 N_1, N_2 分别为类别为 $\mathcal{C}_1\mathcal{C}_2$ 的样本数，N 为样本总数；

与最小二乘法的关系

定理

当目标值用(24)表示时，最小二乘解等价于 Fisher 解。

与最小二乘法的关系

定理

当目标值用(24)表示时，最小二乘解等价于 Fisher 解。

证明

平方和误差函数可表示为

$$E_{\mathcal{D}}(w_0, \boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x}_n + w_0 - t_n\right)^2$$

令 $E(w_0, \boldsymbol{w})$ 关于 w_0 和 $\boldsymbol{w}$ 的导数为零可得

$$\sum_{n=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x}_n + w_0 - t_n\right) = 0, \tag{25}$$

$$\sum_{n=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{x}_n + w_0 - t_n\right)\boldsymbol{x}_n = 0. \tag{26}$$

与最小二乘法的关系

证明

由(24)可知，

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0,$$

再由(25)可得

$$w_0 = -\boldsymbol{w}^T \boldsymbol{m}$$

其中 $\boldsymbol{m}$ 为所有样本的均值，即

$$\boldsymbol{m} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n = \frac{1}{N}(N_1 \boldsymbol{m}_1 + N_2 \boldsymbol{m}_2), \quad \boldsymbol{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \boldsymbol{x}_n, \quad \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \boldsymbol{x}_n,$$

(26)可改写为

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

与最小二乘法的关系

证明

注意到

$$\sum_{n=1}^{N} t_n \boldsymbol{x}_n = \sum_{n\in\mathcal{C}_1} \frac{N}{N_1}\boldsymbol{x}_n - \sum_{n\in\mathcal{C}_2} \frac{N}{N_2}\boldsymbol{x}_n = N(\boldsymbol{m}_1 - \boldsymbol{m}_2),$$

和

$$\begin{aligned}
&\sum_{n=1}^{N} \boldsymbol{x}_n \left(\boldsymbol{x}_n - \boldsymbol{m}\right)^T = \sum_{n\in\mathcal{C}_1} \boldsymbol{x}_n \left(\boldsymbol{x}_n - \boldsymbol{m}\right)^T + \sum_{n\in\mathcal{C}_2} \boldsymbol{x}_n \left(\boldsymbol{x}_n - \boldsymbol{m}\right)^T \\
= \quad & \boldsymbol{S}_W + \sum_{n\in\mathcal{C}_1} \boldsymbol{m}_1(\boldsymbol{m}_1 - \boldsymbol{m})^T + \sum_{n\in\mathcal{C}_2} \boldsymbol{m}_2(\boldsymbol{m}_2 - \boldsymbol{m})^T \\
= \quad & \boldsymbol{S}_W + \frac{N_2}{N}\sum_{n\in\mathcal{C}_1} \boldsymbol{m}_1(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T + \frac{N_1}{N}\sum_{n\in\mathcal{C}_2} \boldsymbol{m}_2(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \\
= \quad & \boldsymbol{S}_W + \frac{N_1 N_2}{N}(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T = \boldsymbol{S}_W + \frac{N_1 N_2}{N}\boldsymbol{S}_B,
\end{aligned}$$

与最小二乘法的关系

证明

(26)可写成

$$\left(\boldsymbol{S}_W + \frac{N_1 N_2}{N}\boldsymbol{S}_B\right)\boldsymbol{w} = N(\boldsymbol{m}_1 - \boldsymbol{m}_2).$$

由于 $\boldsymbol{S}_B\boldsymbol{w} \propto \boldsymbol{m}_1 - \boldsymbol{m}_2$，故

$$\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1).$$

由此可知，由最小二乘法确定的权向量恰好与 Fisher 准则得到的结果一致。

多分类的 Fisher 判别函数

现在考虑 $K > 2$ 分类问题，假设输入空间的维度 $D > K$，引入 $d > 1$ 个线性特征

$$y_k = \boldsymbol{w}_k^T \boldsymbol{x}, \quad k = 1, \cdots, d.$$

记

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_d \end{pmatrix} \in \mathbb{R}^d, \quad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_d \end{pmatrix} \in \mathbb{R}^{D \times d}$$

则

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}.$$

多分类的 Fisher 判别函数

类内协方差矩阵的推广

$$\boldsymbol{S}_W = \sum_{k=1}^{K} \boldsymbol{S}_k \tag{27}$$

其中

$$\boldsymbol{S}_k = \sum_{n \in \mathcal{C}_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^T \tag{28}$$

$$\boldsymbol{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \boldsymbol{x}_n. \tag{29}$$

其中 N_k 表示类 $\mathcal{C}_k$ 中的样本个数。

多分类的 Fisher 判别函数

类间协方差矩阵的推广　考虑整体的协方差矩阵

$$\boldsymbol{S}_T = \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{m})(\boldsymbol{x}_n - \boldsymbol{m})^T, \tag{30}$$

其中 $\boldsymbol{m}$ 表示所有样本的均值，即

$$\boldsymbol{m} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{m}_k.$$

类间协方差矩阵可定义为

$$\begin{aligned} \boldsymbol{S}_B &= \boldsymbol{S}_T - \boldsymbol{S}_W \\ &= \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\boldsymbol{x}_n - \boldsymbol{m})(\boldsymbol{x}_n - \boldsymbol{m})^T - \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x}_n - \boldsymbol{m}_k)^T \\ &= \sum_{k=1}^{K} N_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T. \end{aligned} \tag{31}$$

请注意，这些协方差矩阵是定义在 $\boldsymbol{x}$ 空间的。

多分类的 Fisher 判别函数

类似地，也可以在 d 维 $\boldsymbol{y}$ 空间定义协方差矩阵：

$$\boldsymbol{s}_W = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\boldsymbol{y}_n - \boldsymbol{\mu}_k)(\boldsymbol{y}_n - \boldsymbol{\mu}_k)^T, \tag{32}$$

$$\boldsymbol{s}_B = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T, \tag{33}$$

其中

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \boldsymbol{y}_n, \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k. \tag{34}$$

多分类的 Fisher 判别函数

Fisher 判别的思想仍然是希望极大化一个函数 $J(\boldsymbol{W})$，其中 $\mathcal{J}(\boldsymbol{W})$ 使得类间方差较大而类内方差较小。关于 $\mathcal{J}(\boldsymbol{W})$ 的选择有很多方式，但最常见的一种选择是

$$\mathcal{J}(\boldsymbol{W}) = \mathrm{tr}\left\{\boldsymbol{s}_W^{-1}\boldsymbol{s}_B\right\} = \mathrm{tr}\left\{\left(\boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}\right)^{-1}\left(\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W}\right)\right\}. \tag{35}$$

虽然求优化问题

$$\boldsymbol{W}^* = \arg\max_{\boldsymbol{W}} \mathcal{J}(\boldsymbol{W})$$

有些繁琐，但其推导是直接的。可验证 $\boldsymbol{W}^*$ 由 $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ 的 d 个最大特征值所对应的特征向量确定。

概率判别式模型

概率判别式模型

logistic 回归 (logistic regression)

logistic 回归 (logistic regression)

logistic 回归模型

logistic 回归模型是一种二分类模型，由条件概率分布 $P(y \mid \boldsymbol{x})$ 表示，形式为参数化的 logistic 分布，即

$$P(t=1 \mid \boldsymbol{x}) = y = \sigma(\boldsymbol{w}^T \phi(\boldsymbol{x})), \tag{36}$$

$$P(t=0 \mid \boldsymbol{x}) = 1 - P(t=1 \mid \boldsymbol{x}), \tag{37}$$

其中 $\boldsymbol{x} \in \mathbb{R}^n$ 为输入，$t \in \{0, 1\}$ 为输出，$\sigma(\cdot)$ 为 Sigmoid 函数，即

$$\sigma(x) = \frac{e^x}{1 + e^x}, \tag{38}$$

logistic 回归 (logistic regression)

logistic 回归模型

logistic 回归模型是一种二分类模型，由条件概率分布 $P(y \mid \boldsymbol{x})$ 表示，形式为参数化的 logistic 分布，即

$$P(t=1 \mid \boldsymbol{x}) = y = \sigma(\boldsymbol{w}^T \phi(\boldsymbol{x})), \tag{36}$$

$$P(t=0 \mid \boldsymbol{x}) = 1 - P(t=1 \mid \boldsymbol{x}), \tag{37}$$

其中 $\boldsymbol{x} \in \mathbb{R}^n$ 为输入，$t \in \{0,1\}$ 为输出，$\sigma(\cdot)$ 为 Sigmoid 函数，即

$$\sigma(x) = \frac{e^x}{1+e^x}, \tag{38}$$

对于给定的输入实例 $\boldsymbol{x}$，按照(36)和(37)可以求得 $P(y=1 \mid \boldsymbol{x})$ 和 $P(y=0 \mid \boldsymbol{x})$。logistic 回归比较两个条件概率值的大小，将实例 $\boldsymbol{x}$ 分到概率值较大的那一类。

logistic 回归 (logistic regression)

设 $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^M$ 为基函数构成的向量，表示 $\boldsymbol{x}$ 的 M 维特征，即

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} \phi_0(\boldsymbol{x}) & \phi_1(\boldsymbol{x}) & \cdots & \phi_{M-1}(\boldsymbol{x}) \end{pmatrix}^T, \tag{39}$$

$\boldsymbol{w} \in \mathbb{R}^M$ 为权值向量，刻画的是各特征的重要性，即

$$\boldsymbol{w} = \begin{pmatrix} w_0 & w_1 & \cdots & w_{M-1} \end{pmatrix}^T. \tag{40}$$

logistic 回归 (logistic regression)

Bernouli 分布

若随机变量 t 只取 0 和 1 两个值，并且相应的概率为

$$P(t=1)=\mu, \quad P(t=0)=1-\mu, \quad 0 \leq \mu \leq 1,$$

则称随机变量 t 服从参数为 μ 的伯努利分布。上式也可表示称

$$p(t \mid \mu)=\mu^{t}(1-\mu)^{1-t}.$$

注

Bernouli 分布的期望为

$$\mathbb{E}[t]=\mu,$$

方差为

$$\operatorname{var}[t]=\mu(1-\mu).$$

logistic 回归 (logistic regression)

设有数据集 $\{\boldsymbol{x}_n, t_n\}_{n=1}^N$，其中 $t_n \in \{0, 1\}$, 可通过极大似然法来估计参数 $\boldsymbol{w}$。

似然函数可表示为

$$P(\mathbf{t} \mid \boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}, \tag{41}$$

其中

$$\mathbf{t} = \begin{pmatrix} t_1 & t_2 & \cdots & t_N \end{pmatrix}^T, \quad y_n = \sigma(\boldsymbol{w}^T\boldsymbol{\phi}_n), \quad \boldsymbol{\phi}_n = \boldsymbol{\phi}(\boldsymbol{x}_n).$$

logistic 回归 (logistic regression)

取似然函数负对数，即得交叉熵 (cross-entropy) 误差函数：

Cross-Entropy Error Function

$$
\begin{aligned}
E(\boldsymbol{w}) &= -\ln P(\mathbf{t} \mid \boldsymbol{w}) \\
&= -\sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)] \\
&= -\sum_{n=1}^{N} \left[t_n \ln \sigma(\boldsymbol{w}^T \boldsymbol{\phi}_n) + (1 - t_n) \ln(1 - \sigma(\boldsymbol{w}^T \boldsymbol{\phi}_n)) \right].
\end{aligned} \tag{42}
$$

$E(\boldsymbol{w})$ 的梯度

记

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi}_1^T \\ \vdots \\ \boldsymbol{\phi}_N^T \end{pmatrix} \in \mathbb{R}^{N \times M}, \tag{43}$$

则 $E(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的梯度为

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n = \boldsymbol{\Phi}^T(\boldsymbol{y} - \mathbf{t}) \tag{44}$$

证明

注意到 sigmoid 函数的性质 $\sigma' = \sigma(1-\sigma)$, 令 $z = \boldsymbol{w}^T\phi$, 则有

$$\frac{\partial \ln \sigma(\boldsymbol{w}^T\phi)}{\partial \boldsymbol{w}} = \frac{\partial \ln \sigma}{\sigma} \cdot \frac{\partial \sigma(z)}{\partial z} \cdot \frac{\partial \boldsymbol{w}^T\phi}{\partial \boldsymbol{w}} = (1-\sigma)\phi$$

同理,

$$\frac{\partial \ln(1-\sigma(\boldsymbol{w}^T\phi))}{\partial \boldsymbol{w}} = \sigma\phi$$

因此

$$\begin{aligned}
\partial_{\boldsymbol{w}} E(\boldsymbol{w}) &= -\sum_{n=1}^{N}\left[t_n \frac{\partial \ln \sigma(\boldsymbol{w}^T\phi_n)}{\partial \boldsymbol{w}} + (1-t_n)\frac{\partial \ln(1-\sigma(\boldsymbol{w}^T\phi_n))}{\partial \boldsymbol{w}}\right] \\
&= -\sum_{n=1}^{N}(t_n - \sigma(\boldsymbol{w}^T\phi(\boldsymbol{x}_n)))\phi_n \\
&= \sum_{n=1}^{N}(y_n - t_n)\phi_n = \boldsymbol{\Phi}^T(\boldsymbol{y}-\mathbf{t}).
\end{aligned}$$

logistic 回归 (logistic regression)

对于 Logistic 回归，极大似然估计不再有解析解。这里介绍求最小值问题的一种迭代格式，它基于 Newton-Raphson 迭代方法：

Newton-Raphson 迭代

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

其中 $\boldsymbol{H}$ 为 $E(\boldsymbol{w}) \in \mathbb{R}^{M\times M}$ 的 Hessian 矩阵，其元素为

$$h_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j}. \tag{45}$$

Newton-Raphson 迭代用于线性回归模型

设线性回归模型 $y(x) = \boldsymbol{w}^T \phi(\boldsymbol{x})$, 其误差函数为

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} \left(t_n - \boldsymbol{w}^T \phi(\boldsymbol{x}_n)\right)^2,$$

它关于 $\boldsymbol{w}$ 的梯度和 Hessian 矩阵为

$$\nabla E(\boldsymbol{w}) = \boldsymbol{\Phi}^T(\boldsymbol{y} - \mathbf{t}), \tag{46}$$

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \sum_{n=1}^{T} \phi_n \phi_n^T = \boldsymbol{\Phi}^T \boldsymbol{\Phi}. \tag{47}$$

logistic 回归 (logistic regression)

于是, Newton-Raphson 迭代格式为

$$\begin{aligned} \boldsymbol{w}^{new} &= \boldsymbol{w}^{old} - \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^{old} - \boldsymbol{\Phi}^T\mathbf{t}\right) \\ &= \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\mathbf{t}, \end{aligned} \tag{48}$$

此即标准的最小二乘解。

注

此时的误差函数是二次的，因此 Newton-Raphson 公式迭代一次就给出了精确解。

logistic 回归 (logistic regression)

以下讨论如何将 Newton-Raphson 迭代法应用于 logistic 回归模型上。

logistic 回归 (logistic regression)

以下讨论如何将 Newton-Raphson 迭代法应用于 logistic 回归模型上。

$E(\boldsymbol{w})$ 的 Hessian 矩阵

$E(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的 Hessian 矩阵为

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T = \boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}, \tag{49}$$

其中 $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$ 为对角阵，其形式为

$$\boldsymbol{\Lambda} = \begin{pmatrix} y_1(1-y_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_N(1-y_N) \end{pmatrix}.$$

logistic 回归 (logistic regression)

以下讨论如何将 Newton-Raphson 迭代法应用于 logistic 回归模型上。

$E(\boldsymbol{w})$ 的 Hessian 矩阵

$E(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的 Hessian 矩阵为

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T = \boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}, \tag{49}$$

其中 $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$ 为对角阵，其形式为

$$\boldsymbol{\Lambda} = \begin{pmatrix} y_1(1-y_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_N(1-y_N) \end{pmatrix}.$$

注

$E(\boldsymbol{w})$ 的 Hessian 矩阵 $\boldsymbol{H}$ 不再是常量，它依赖于 $\boldsymbol{w}$。

logistic 回归 (logistic regression)

定理

误差函数 $E(\boldsymbol{w})$ 是关于 $\boldsymbol{w}$ 的凸函数，从而存在唯一的最小值点。

logistic 回归 (logistic regression)

定理

误差函数 $E(\boldsymbol{w})$ 是关于 $\boldsymbol{w}$ 的凸函数，从而存在唯一的最小值点。

证明

由于 $0 < y_n < 1$，对任意的 $\boldsymbol{u} \in \mathbb{R}^N$ 恒成立

$$\boldsymbol{u}^T \boldsymbol{H} \boldsymbol{u} = (\boldsymbol{\Phi} \boldsymbol{u})^T \boldsymbol{\Lambda} (\boldsymbol{\Phi} \boldsymbol{u}) > 0,$$

因此 $\boldsymbol{H}$ 是正定的，于是 $E(\boldsymbol{w})$ 是关于 $\boldsymbol{w}$ 的一个凸函数，从而有唯一的最小值。

logistic 回归 (logistic regression)

Newton-Raphson 迭代格式用于 logistic 回归模型

$$
\begin{aligned}
\boldsymbol{w}^{new} &= \boldsymbol{w}^{old} - \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T(\boldsymbol{y}-\mathbf{t}) \\
&= \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\left[\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{w}^{old} - \boldsymbol{\Phi}^T(\boldsymbol{y}-\mathbf{t})\right] \qquad (50) \\
&= \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{z}
\end{aligned}
$$

其中

$$
\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{w}^{old} - \boldsymbol{\Lambda}^{-1}(\boldsymbol{y}-\mathbf{t}). \qquad (51)
$$

logistic 回归 (logistic regression)

Newton-Raphson 迭代格式用于 logistic 回归模型

$$\begin{aligned} \boldsymbol{w}^{new} &= \boldsymbol{w}^{old} - \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T(\boldsymbol{y}-\mathbf{t}) \\ &= \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\left[\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{w}^{old} - \boldsymbol{\Phi}^T(\boldsymbol{y}-\mathbf{t})\right] \qquad (50) \\ &= \left(\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Lambda}\boldsymbol{z} \end{aligned}$$

其中

$$\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{w}^{old} - \boldsymbol{\Lambda}^{-1}(\boldsymbol{y}-\mathbf{t}). \qquad (51)$$

注

由(50)可知，它实际是加权最小二乘的法方程组。由于 $\boldsymbol{\Lambda}$ 依赖于 $\boldsymbol{w}$，必须迭代地求解法方程组，每次使用新的 $\boldsymbol{w}$ 来计算修正的 $\boldsymbol{\Lambda}$，于是该算法被称为迭代重加权最小二乘 (iterate reweighted least squares, IRLS)。

概率判别式模型

softmax 回归

softmax 回归

softmax 回归主要用于多分类问题中。

softmax 函数

设有向量

$$\boldsymbol{a} = (a_1, \cdots, a_K)^T \in \mathbb{R}^K,$$

softmax 函数是一个 $\mathbb{R}^K \to \mathbb{R}^K$ 的函数，记为 $\boldsymbol{s}(\boldsymbol{a})$，其元素为

$$s_i = s_i(\boldsymbol{a}) = \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}}, \quad i = 1, \cdots, K. \tag{52}$$

softmax 函数的导数

softmax 函数的导数是一个 Jacobi 矩阵，即

$$\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{a}} = \begin{pmatrix} \partial_{a_1} s_1 & \cdots & \partial_{a_K} s_1 \\ \vdots & \ddots & \vdots \\ \partial_{a_1} s_K & \cdots & \partial_{a_K} s_K \end{pmatrix} \tag{53}$$

其中

$$\partial_{a_j} s_i = \frac{\partial s_i}{\partial a_j} = \begin{cases} s_i(1 - s_i), & i = j \\ -s_i s_j, & i \neq j \end{cases} = s_i(\delta_{ij} - s_j),$$

δ_{ij} 为 Kronecker 函数，即

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

证明

记 $\Sigma = \sum_{k=1}^{K} e^{a_k}$，则 s_i 可表示为

$$s_i = \frac{e^{a_i}}{\Sigma}.$$

以下分两种情形计算 $\partial_{a_j} s_i$：

- 当 $i = j$ 时，

$$\partial_{a_i} s_i = \frac{\partial}{\partial a_i} \frac{e^{a_i}}{\Sigma} = \frac{e^{a_i}\Sigma - e^{a_i} e^{a_i}}{\Sigma^2} = \frac{e^{a_i}}{\Sigma} \frac{\Sigma - e^{a_i}}{\Sigma} = s_i(1 - s_i).$$

- 当 $i \neq j$ 时，

$$\partial_{a_j} s_i = \frac{\partial}{\partial a_j} \frac{e^{a_i}}{\Sigma} = \frac{-e^{a_i} e^{a_j}}{\Sigma^2} = -\frac{e^{a_i}}{\Sigma} \frac{e^{a_j}}{\Sigma} = -s_i s_j.$$

综上所述，结论得证。

softmax 回归

类别 $\mathcal{C}_k$ 的条件概率可表示为

$$P(t_k \mid \boldsymbol{x}) = y_k = s_k(\boldsymbol{a}), \tag{54}$$

其中

$$\boldsymbol{a} = (a_1, \cdots, a_K)^T, \quad a_k = \boldsymbol{w}_k^T \phi(\boldsymbol{x}). \tag{55}$$

softmax 回归模型

定义

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{w}_1^T \\ \vdots \\ \boldsymbol{w}_K^T \end{pmatrix} \in \mathbb{R}^{K \times M}, \tag{56}$$

则 softmax 回归模型可表示为

$$\boldsymbol{y} = \boldsymbol{s}(\boldsymbol{W}\phi(\boldsymbol{x})) \in \mathbb{R}^K. \tag{57}$$

softmax 回归

多分类问题的 softmax 回归

给定训练数据 $\{\boldsymbol{x}_n, \boldsymbol{t}_n\}$，确定 softmax 回归模型(57)的参数 $\boldsymbol{W}$。一般地，目标向量 $\boldsymbol{t}_n$ 以"one hot ecoding" 的形式表示，即若样本 $\boldsymbol{x}_n$ 属于类 $\mathcal{C}_k$，则

$$\boldsymbol{t}_n = \boldsymbol{e}_k \in \mathbb{R}^K.$$

多项式分布 (multinouli distribution)

多项式分布可看做是 Bernouli 分布对于多个输出的一个推广。设有随机变量 $\boldsymbol{t} \in \mathbb{R}^K$，每种状态的取值可能性只有 0 或 1，且对应的概率为

$$P(t_k = 1) = \mu_k, \quad 0 \leq \mu_k \leq 1, \quad 1 \leq \mu_k \leq 1,$$

则称随机变量 $\boldsymbol{t}$ 服从参数为 $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_K)^T$ 的多项式分布。上式也可表示成

$$p(\boldsymbol{t} \mid \boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{t_k}$$

softmax 回归

此时，似然函数可写成

$$P(\boldsymbol{T} \mid \boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} P(\boldsymbol{t}_k \mid \boldsymbol{x}_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}, \tag{58}$$

其中 $y_{nk} = s_k(\phi(\boldsymbol{x}_n))$，

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{pmatrix} \in \mathbb{R}^{N \times K}.$$

对其取负对数即得

多分类的交叉熵 (cross-entropy) 误差函数

$$
\begin{aligned}
E(\boldsymbol{W}) &= E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) \\
&= -\ln P(\boldsymbol{T} \mid \boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) \\
&= -\sum_{n=1}^{N}\sum_{i=1}^{K} t_{ni} \ln y_{ni} \\
&= -\sum_{n=1}^{N} (\ln \boldsymbol{y}_n)^T \boldsymbol{t}_n
\end{aligned}
\tag{59}
$$

$E(\boldsymbol{W})$ 的梯度

对 $k = 1, \cdots, K$,

$$\nabla_{\boldsymbol{w}_k} E(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K) = \sum_{n=1}^{N} (y_{nk} - t_{nk}) \boldsymbol{\phi}_n = \boldsymbol{\Phi}^T \begin{pmatrix} y_{1k} \\ \vdots \\ y_{Nk} \end{pmatrix}, \quad (60)$$

即

$$\nabla_{\boldsymbol{W}} E(\boldsymbol{W}) = \boldsymbol{\Phi}^T (\boldsymbol{Y} - \boldsymbol{T}), \quad (61)$$

其中

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \end{pmatrix} \in \mathbb{R}^{N \times K}, \quad \boldsymbol{T} = \begin{pmatrix} \boldsymbol{t}_1 \\ \vdots \\ \boldsymbol{t}_N \end{pmatrix} \in \mathbb{R}^{N \times K}$$

证明

由链式法则知

$$\frac{\partial \ln \boldsymbol{y}_n}{\partial \boldsymbol{w}_k} = \frac{\partial \ln \boldsymbol{s}_n}{\partial \boldsymbol{w}_k} = \frac{\partial \ln \boldsymbol{s}_n}{\partial \boldsymbol{s}_n} \frac{\partial \boldsymbol{s}(\boldsymbol{a}_n)}{\partial \boldsymbol{a}_n} \frac{\partial \boldsymbol{a}_n}{\partial \boldsymbol{w}_k}.$$

注意到

$$\frac{\partial \ln \boldsymbol{s}_n}{\partial \boldsymbol{s}_n} = \begin{pmatrix} s_{n1}^{-1} & 0 & \cdots & 0 \\ 0 & s_{n2}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{nK}^{-1} \end{pmatrix} \in \mathbb{R}^{K \times K}$$

$$\frac{\partial \boldsymbol{s}(\boldsymbol{a}_n)}{\partial \boldsymbol{a}_n} = \begin{pmatrix} s_{n1}(1 - s_{n1}) & -s_{n1}s_{n2} & \cdots & -s_{n1}s_{nK} \\ -s_{n2}s_{n1} & s_{n2}(1 - s_{n2}) & \cdots & -s_{n2}s_{nK} \\ \vdots & \vdots & \ddots & \vdots \\ -s_{nK}s_{n1} & s_{nK}s_{n2} & \cdots & s_{nK}(1 - s_{nK}) \end{pmatrix} \in \mathbb{R}^{K \times K}$$

证明 (续)

$$\frac{\partial \boldsymbol{a}_n}{\partial \boldsymbol{w}_k} = \frac{\partial(\boldsymbol{W}\boldsymbol{\phi}_n)}{\partial \boldsymbol{w}_k} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \boldsymbol{\phi}_n^T \\ \vdots \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{K \times M}$$

以及

$$\sum_{k=1}^{K} t_{nk} = 1,$$

证明 (续)

从而有

$$\left(\frac{\partial \ln \boldsymbol{s}_n}{\partial \boldsymbol{w}_k}\right)^T \boldsymbol{t}_n = \begin{pmatrix} \mathbf{0} & \cdots & \boldsymbol{\phi}_n & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} (1-s_{n1}) & -s_{n1} & \cdots & -s_{n1} \\ -s_{n2} & (1-s_{n2}) & \cdots & -s_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ -s_{nK} & s_{nK} & \cdots & (1-s_{nK}) \end{pmatrix} \begin{pmatrix} t_{n1} \\ t_{n2} \\ \vdots \\ t_{nK} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{0} & \cdots & \boldsymbol{\phi}_n & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} t_{n1} - s_{n1} \sum_{k=1}^{K} t_{nk} \\ t_{n2} - s_{n2} \sum_{k=1}^{K} t_{nk} \\ \vdots \\ t_{nK} - s_{nK} \sum_{k=1}^{K} t_{nk} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{0} & \cdots & \boldsymbol{\phi}_n & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} t_{n1} - s_{n1} \\ t_{n2} - s_{n2} \\ \vdots \\ t_{nK} - s_{nK} \end{pmatrix}$$

$$= (t_{nk} - s_{nk})\boldsymbol{\phi}_n$$

证明 (续)

于是，

$$\frac{\partial E}{\partial \boldsymbol{w}_k} = -\sum_{n=1}^{N} \left(\frac{\partial \ln \boldsymbol{s}_n}{\partial \boldsymbol{w}_k} \right)^T \boldsymbol{t}_n = \sum_{n=1}^{N} (y_{nk} - t_{nk}) \boldsymbol{\phi}_n.$$

证毕.