1.

1) 最小错误率决策：应当寻找最大的后验概率，来使错误率最小.

$$P(error|x) = 1 - P(wi|x) \quad if\ you\ decide\ wi$$

- Minimum error decision: Maximum a posteriori (MAP)

$$\text{Decide } \omega_i \text{ if } P(\omega_i|\mathrm{x}) > P(\omega_j|\mathrm{x}) \qquad \text{for all } j \neq i$$

最小风险决策:通过计算不同类别的最小风险决策,选择最小的 R(αi|x)所在的类别.

贝叶斯最小风险决策：

Condition risk

$$R(\alpha_i|\mathrm{x}) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j)P(\omega_j|\mathrm{x})$$

Minimum risk decision (Bayes decision)

$$\arg\min_i R(\alpha_i \mid x)$$

2)

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} \mathrm{x}_k$$

结论记住

$$\widehat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n} (\mathrm{x}_k - \hat{\mu})(\mathrm{x}_k - \hat{\mu})^t$$

$$\boldsymbol{u1} = (3,6)^T$$

$$\boldsymbol{\Sigma 1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\boldsymbol{u2} = (3,-2)^T$$

$$\boldsymbol{\Sigma 2} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Case 3: Σ_i= arbitrary

$$g_i(\mathrm{x}) = -\frac{1}{2}(\mathrm{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathrm{x} - \boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln |\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$g_i(\mathrm{x}) = \mathrm{x}^t \mathbf{W}_i \mathrm{x} + \mathrm{w}_i^t \mathrm{x} + w_{i0}$$

$$\mathbf{W}_i = -\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} \qquad \mathrm{w}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$$

$$w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^t \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln |\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$g_1(x) = x^t W_1 x + w_1^t x + w_{10}$$

$$W_1 = -\frac{1}{2}\Sigma_1^{-1} = -\frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -\frac{1}{4} \end{pmatrix}$$

$$w_1 = \Sigma_1^{-1} u_1 = \begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 3 \\ 6 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \end{pmatrix}$$

$$w_{10} = -\frac{1}{2} u_1^t \Sigma_1^{-1} u_1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(w_1)$$

$$= -\frac{1}{2}(3\ 6)\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 3 \\ 6 \end{pmatrix} - \frac{1}{2}\ln 1 + \ln P(w_1)$$

$$= -\frac{1}{2}(6\ 3)\begin{pmatrix} 3 \\ 6 \end{pmatrix} + \ln P(w_1)$$

$$= \cancel{-9}\ {-18} + \ln P(w_1)$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$g_1(x) = (x_1\ x_2)\begin{pmatrix} -1 & 0 \\ 0 & -\frac{1}{4} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 6 \\ 3 \end{pmatrix}^T\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \cancel{-9}\ {-18} + \ln P(w_1)$$

$$= -x_1^2 - \frac{1}{4}x_2^2 + 6x_1 + 3x_2 + \cancel{-9}\ {-18}.$$

同理.

$$g_2(x) = x^t W_2 x + w_2^t x + w_{20}.$$

$$W_2 = -\frac{1}{2}\Sigma_2^{-1} = -\frac{1}{2}\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} -\frac{1}{4} & 0 \\ 0 & -\frac{1}{4} \end{pmatrix}$$

$$w_2 = \Sigma_2^{-1} u_2 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 3 \\ -2 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ -1 \end{pmatrix}$$

$$P(w_1) = P(w_2)$$

$$w_{20} = -\frac{1}{2} u_2^t \Sigma_2^{-1} u_2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(w_2)$$

$$= -\frac{1}{2}(3\ {-2})\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 3 \\ -2 \end{pmatrix} - \frac{1}{2}\ln 4 + \ln P(w_2)$$

$$= -\frac{13}{4} - \ln 2 + \ln P(w_2)$$

$$g_2(x) = -\frac{1}{4}x_1^2 - \frac{1}{4}x_2^2 + \frac{3}{2}x_1 - x_2 - \frac{13}{4} - \ln 2$$

令 $g_1(x) = g_2(x)$, 则

$$x_2 = 3.514 - 1.125x_1 + 0.1875x_1^2$$

2.

1)

Case 2: $\Sigma_i = \Sigma$

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$\Rightarrow \quad g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln P(\omega_i)$$

– 展开二次式 $(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$

线性判别函数！ $g_i(\mathbf{x}) = \mathbf{w}_i^t\mathbf{x} + w_{i0}$

$$\mathbf{w}_i = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i \qquad w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

Case 3: Σ_i= arbitrary

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$g_i(\mathbf{x}) = \mathbf{x}^t\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^t\mathbf{x} + w_{i0}$$

$$\mathbf{W}_i = -\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} \qquad \mathbf{w}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$$

$$w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^t\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

2)

– 二类决策面 $g_i(\mathbf{x}) = g_j(\mathbf{x})$

协方差任意，不一定垂直

$$\Rightarrow \quad \mathbf{w}^t(\mathbf{x}-\mathbf{x}_0) = 0 \qquad \mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \frac{\ln[P(\omega_i)/P(\omega_j)]}{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

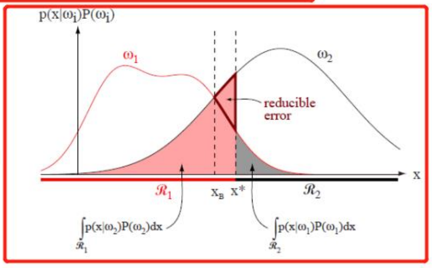
概率是体积，概率密度对应……

- 2类的情况

$$\boxed{\begin{aligned} P(error) &= P(\mathbf{x}\in\mathcal{R}_2,\omega_1) + P(\mathbf{x}\in\mathcal{R}_1,\omega_2) \\ &= P(\mathbf{x}\in\mathcal{R}_2|\omega_1)P(\omega_1) + P(\mathbf{x}\in\mathcal{R}_1|\omega_2)P(\omega_2) \\ &= \int_{\mathcal{R}_2} p(\mathbf{x}|\omega_1)P(\omega_1)\,d\mathbf{x} + \int_{\mathcal{R}_1} p(\mathbf{x}|\omega_2)P(\omega_2)\,d\mathbf{x}. \end{aligned}}$$

- 一般情况

$$\begin{aligned} P(correct) &= \sum_{i=1}^{c} P(\mathbf{x}\in\mathcal{R}_i,\omega_i) \\ &= \sum_{i=1}^{c} P(\mathbf{x}\in\mathcal{R}_i|\omega_i)P(\omega_i) \\ &= \sum_{i=1}^{c}\int_{\mathcal{R}_i} p(\mathbf{x}|\omega_i)P(\omega_i)\,d\mathbf{x} \end{aligned}$$

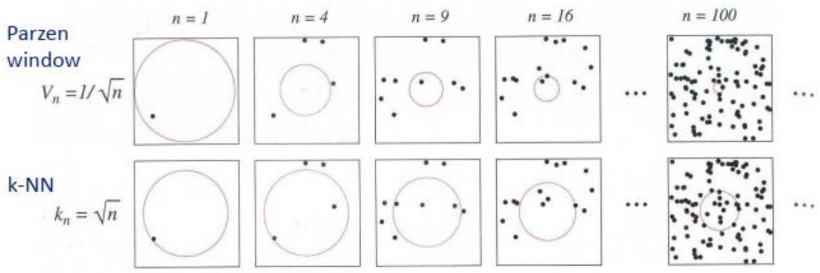


决策面为x_B时为最小错误率分类

3.

1) Parzen 窗与 K 紧邻估计的区别

- 非参数概率密度估计
  - Parzen window: 固定局部区域体积V，k变化
  - k-nearest neighbor: 固定局部样本数k，V变化



2)

$$k = \sum_{i=1}^{c} k_i, \quad p_n(\mathbf{x}, \omega_i) = \frac{k_i}{nV}$$

$$p_n(\omega_i \mid \mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{i=1}^{c} p_n(\mathbf{x}, \omega_i)} = \frac{k_i}{k}, \qquad \omega_m = \arg\max_i \{p_n(\omega_i \mid \mathbf{x})\}$$

这就是K近邻分类器！

4.

1)

$y = [x^T, 1]^T \in R^{d+1}$，属于$\omega_2$类的$y$变成$-y$.

$y_1 = [3, 2, 1]^T$，$y_2 = [2, 2, 1]^T$，$y_3 = [-1, -0, -1]^T$，$y_4 = [-2, 0, -1]^T$

$$\boxed{\begin{gathered} a_{k+1} = a_k + \eta_k \sum_{y \in Y_k} y \\ Y_k \text{ 为 } a_k \text{ 错分样本集合} \end{gathered}}$$

① 初始权向量 $a_1 = (1, 0, 0)^T$

$$\left.\begin{aligned}
a_1^T y_1 &= (1\ 0\ 0)(3\ 2\ 1)^T = 3 > 0. \\
a_1^T y_2 &= (1\ 0\ 0)(2\ 2\ 1)^T = 2 > 0 \\
a_1^T y_3 &= (1\ 0\ 0)(-1\ 0\ -1)^T = -1 < 0 \\
a_1^T y_4 &= (1\ 0\ 0)(-2\ 0\ -1)^T = -2 < 0
\end{aligned}\right\} \quad Y_{(1)} = \{y_3, y_4\}$$

② 更新 $a_2 = a_1 + y_3 + y_4 = [1\ 0\ 0]^T + (-1\ 0\ -1)^T + (-2, 0\ -1)^T = (-2\ 0\ -2)^T$

$$\left.\begin{aligned}
a_2^T y_1 &= (-2\ 0\ -2)(3\ 2\ 1)^T = -8 < 0. \\
a_2^T y_2 &= (-2\ 0\ -2)(2\ 2\ 1)^T = -6 < 0 \\
a_2^T y_3 &= (-2\ 0\ -2)(-1\ 0\ -1)^T = 4 > 0. \\
a_2^T y_4 &= (-2\ 0\ -2)(-2\ 0\ -1)^T = 6 > 0
\end{aligned}\right\} \quad Y_{(2)} = \{y_1, y_2\}$$

③ 更新 $a_3 = a_2 + y_1 + y_2 = (-2\ 0\ -2)^T + (3\ 2\ 1)^T + (2\ 2\ 1)^T = (3\ 4\ 0)^T$

$$\left.\begin{aligned}
a_3^T y_1 &= (3\ 4\ 0)(3\ 2\ 1)^T = 17 > 0 \\
a_3^T y_2 &= (3\ 4\ 0)(2\ 2\ 1)^T = 14 > 0 \\
a_3^T y_3 &= (3\ 4\ 0)(-1\ 0\ -1)^T = -3 < 0 \\
a_3^T y_4 &= (3\ 4\ 0)(-2\ 0\ -1)^T = -6 < 0
\end{aligned}\right\} \quad Y(3) = \{y_3, y_4\}$$

④ 更新 $a_4 = a_3 + y_3 + y_4 = (3\ 4\ 0)^T + (-1\ 0\ -1)^T + (-2\ 0\ -1)^T = (0\ 4\ -2)^T$.

$$\left.\begin{aligned}
a_4^T y_1 &= (0\ 4\ -2)(3\ 2\ 1)^T = 6 > 0 \\
a_4^T y_2 &= (0\ 4\ -2)(2\ 2\ 1)^T = 6 > 0 \\
a_4^T y_3 &= (0\ 4\ -2)(-1\ 0\ -1)^T = 2 > 0. \\
a_4^T y_4 &= (0\ 4\ -2)(-2\ 0\ -1)^T = 2 > 0.
\end{aligned}\right\} \quad \cancel{Y(4) = \{y_3, y_4\}}.$$

故 $a = (0\ 4\ -2)^T$

5.

1)

5. ① $m_1^{(0)}=(-2,0)^T$, $m_2^{(0)}=(4,0)^T$

$x_1=(-4,1)^T$, $x_2=(-2,1)^T$
$x_3=(-4,-1)^T$, $x_4=(-2,-1)^T$
$x_5=(6,1)^T$, $x_6=(4,1)^T$
$x_7=(6,-1)^T$, $x_8=(4,-1)^T$

x_1 : $d[x_1,m_1^{(0)}]=\sqrt{5}$ ✓
$d[x_1,m_2^{(0)}]=\sqrt{65}$

x_2 : $d[x_2,m_1^{(0)}]=1$ ✓ , $d[x_2,m_2^{(0)}]=\sqrt{5}$

x_3 : $d[x_3,m_1^{(0)}]=\sqrt{5}$ ✓ , $d[x_3,m_2^{(0)}]=\sqrt{65}$

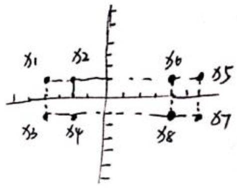
x_4 : $d[x_4,m_1^{(0)}]=1$ ✓ , $d[x_4,m_2^{(0)}]=\sqrt{37}$

x_5 : $d[x_5,m_1^{(0)}]=\sqrt{65}$, $d[x_5,m_2^{(0)}]=\sqrt{5}$ ✓

x_6 : $d[x_6,m_1^{(0)}]=\sqrt{37}$, $d[x_6,m_2^{(0)}]=1$ ✓

x_7 : $d[x_7,m_1^{(0)}]=\sqrt{65}$, $d[x_7,m_2^{(0)}]=\sqrt{5}$ ✓

x_8 : $d[x_8,m_1^{(0)}]=\sqrt{37}$, $d[x_8,m_2^{(0)}]=1$ ✓



故 $G_1^{(1)}=\{x_1,x_2,x_3,x_4\}$ $G_2^{(1)}=\{x_5,x_6,x_7,x_8\}$.

② ~~m_2~~ $m_1^{(1)}=(-3,0)^T$, $m_2^{(1)}=(5,0)^T$

x_1 : $d[x_1,m_1^{(1)}]=\sqrt{2}$ ✓ $d[x_1,m_2^{(1)}]=\sqrt{82}$

x_2 : $d[x_2,m_1^{(1)}]=\sqrt{2}$ ✓ $d[x_2,m_2^{(1)}]=\sqrt{50}$

x_3 : $d[x_3,m_1^{(1)}]=\sqrt{2}$ ✓ $d[x_3,m_2^{(1)}]=\sqrt{82}$

x_4 : $d[x_4,m_1^{(1)}]=\sqrt{2}$ ✓ $d[x_4,m_2^{(1)}]=\sqrt{50}$.

x_5 : $d[x_5,m_1^{(1)}]=\sqrt{82}$ $d[x_5,m_2^{(1)}]=\sqrt{2}$ ✓

x_6 : $d[x_6,m_1^{(1)}]=\sqrt{50}$, $d[x_6,m_2^{(1)}]=\sqrt{2}$ ✓

x_7 : $d[x_7,m_1^{(1)}]=\sqrt{82}$, $d[x_7,m_2^{(1)}]=\sqrt{2}$ ✓

x_8 : $d[x_8,m_1^{(1)}]=\sqrt{50}$ $d[x_8,m_2^{(1)}]=\sqrt{2}$ ✓

$m_2^{(1)}=(-3,0)^T$, $m_2^{(1)}=(5,0)^T$

故分类不再变化，迭代停止。

2)

得到一些经典的算法。其中之一是著名的 **K-均值聚类**算法。引入如下假设：

- 各类出现的先验概率均相等；
- 每个均本点以概率为1属于一个类（后验概率0-1近似）；
  - 计算数据点到类中心的欧氏距离的平方，即计算 $\|\mathbf{x}_k-\hat{\boldsymbol{\mu}}_i\|^2$ ，寻找与样本 $\mathbf{x}_k$ 最近的类中心点，将 $\mathbf{x}_k$ 分给最近的类（即假定协方差矩阵为分无限小单位阵）：

$$\hat{P}(\omega_i\mid\mathbf{x}_k,\hat{\boldsymbol{\theta}})\approx\begin{cases}1, & \text{if } \mathbf{x}_k \text{ is nearest to the center } \hat{\boldsymbol{\mu}}_i\\ 0, & \text{otherwise}\end{cases}$$

6.

1) 原理:

6.4.2 BP算法

- 误差反向传播训练算法
  - 属于监督学习算法，**通过调节各层的权重，使网络学会**由“输入-输出对”组成的训练组。
  - BP算法核心是**梯度下降法**。
  - 权重先从输出层开始修正，再依次修正各层权重
    - 首先修正：**“输出层至最后一个隐含层”**的连接权重
    - 再修正：**“最后一个隐含层至倒数第二个隐含层”**的连接权重，....
    - 最后修正：**“第一隐含层至输入层”**的连接权重。

学习的本质：对网络各连接权重作动态调整！

公式:

- 隐含层—输出层：第 k 个训练样本对权重 w_{hj} 的贡献

$h \rightarrow j$, for sample k:

δ规则： $$\boxed{\Delta w_{hj}\,|_{\text{sample }k} = \eta \delta_j^k y_h^k}$$

权重所联边的起始结点（隐含结点 h）的输出

权重所联边的指向结点（输出结点 j）收集到的误差信号

$$\delta_j^k = f'(net_j^k)\Delta_j^k, \quad \Delta_j^k = t_j^k - z_j^k$$

误差**在权重所联边的指向结点处**计算。

误差大小等于：该结点收集到的误差**乘以**激励函数对“该结点加权和”的导数。

输入-隐层：第 k 个训练样本对权重 w_{ih} 的贡献

$i \rightarrow h$, for sample k:

δ规则： $$\boxed{\Delta w_{ih}\,|_{\text{sample }k} = \eta \delta_h^k x_i^k}$$

w_{ih}所连接的边的起始结点（输入层结点 i）的输出(此时即为样本第 i 个分量)

w_{ih}所连接的边的指向结点（隐含结点 h）收集到的误差信号

2)

2. 自组织映射网络的构造原理：自组织特征映射网中竞争机制的生物学基础是神经元的有序排列以及对外界信息的连续映象。该算法的原理是通过自动寻找样本中的内在规律和本质属性，自组织、自适应地改变网络参数与结构。

学习步骤：

S1：网络初始化—通常采用随机初始化方法

S2：输入向量

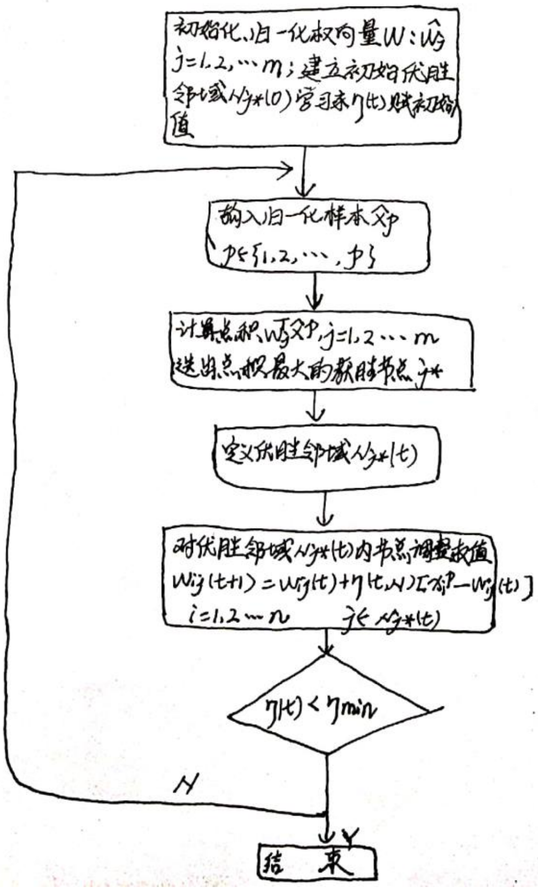
S3：计算映射层的权重向量与输入向量的距离 $d_j=\sqrt{\sum_{i=1}^{n}(x_i-w_{ij})^2}$

S4：选择与权重向量的距离最小的神经元（确定胜者）

S5：调整权重 $\Delta w_{ij}=\eta h(j,j^*)(x_i-w_{ij})$

$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}$

S6：检查是否达到预先设定的要求



7.

1) 简述 LDA 的主要思想

2) 基于上述思想，写两类问题的 LDA 目标函数

3) 最优化上述目标函数，得到 LDA 结果

解：

1) LDA 主要思想：线性判别分析，是从更利于分类的角度的有监督（训练样本有标签）的降维方法。希望数据投影后类内方差最小，类间方差最大。

适用场景：侧重于分类。

2) 两类问题的 LDA 目标函数

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

分子越大:类间越大

分母越小：类内方差小

$$\boxed{(\tilde{\mu_1} - \tilde{\mu_2})^2} = \mathbf{w}^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\mathbf{w} \propto \boxed{\mathbf{w}^T S_b \mathbf{w}}$$ Between class scatter

$$\tilde{\sigma_i^2} = E[(y - \tilde{\mu_i})^2] = E[\mathbf{w}^T(x - \mu)(x - \mu_i)^T\mathbf{w})] = \mathbf{w}^\mathbf{T}\mathbf{\Sigma_i}\mathbf{w}$$ Covariance matrix

$$\boxed{\tilde{\sigma_1}^2 + \tilde{\sigma_2}^2} \propto \boxed{\mathbf{w}^T S_w \mathbf{w}}$$ Within class scatter

最大化下列目标函数

$$FDR(\mathbf{w}) = \frac{\mathbf{w}^T S_b \mathbf{w}}{\mathbf{w}^T S_w \mathbf{w}}$$

Sb：类间离散度

Sw：类内离散度

3) 最大化分子，把分母等于 1 作为约束条件，写出拉格朗日乘子法求出 w 的值。

Our Goal: Find w maximizing FDR(w)

- achieved if w chosen such that:

$$S_b \mathbf{w} = \lambda S_w \mathbf{w}$$

- where lambda is the largest eigenvalue of $S_w^{-1} S_b$
- For two classes, to get the direction of w, use:

$$\mathbf{w} = S_w^{-1}(\mu_1 - \mu_2)$$

方法补充理解：

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

第 i 类示例的集合 X_i

第 i 类示例的均值向量 $\boldsymbol{\mu}_i$

第 i 类示例的协方差矩阵 $\mathbf{\Sigma}_i$

两类样本的中心在直线上的投影：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$

两类样本的协方差：$\boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_1\boldsymbol{w}$

同类样例的投影点尽可能接近 → $\boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_1\boldsymbol{w}$ 尽可能小

异类样例的投影点尽可能远离 → $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2$ 尽可能大

于是，最大化

$$J = \frac{\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\mathbf{\Sigma}_1\boldsymbol{w}} = \frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)\boldsymbol{w}}$$

类内散度矩阵 (within-class scatter matrix)

$$\begin{aligned}\mathbf{S}_w &= \mathbf{\Sigma}_0 + \mathbf{\Sigma}_1 \\ &= \sum_{\boldsymbol{x} \in X_0}(\boldsymbol{x} - \boldsymbol{\mu}_0)(\boldsymbol{x} - \boldsymbol{\mu}_0)^{\mathrm{T}} + \sum_{\boldsymbol{x} \in X_1}(\boldsymbol{x} - \boldsymbol{\mu}_1)(\boldsymbol{x} - \boldsymbol{\mu}_1)^{\mathrm{T}}\end{aligned}$$

类间散度矩阵 (between-class scatter matrix)

$$\mathbf{S}_b = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}$$

令 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$，最大化广义瑞利商等价形式为

$$\begin{aligned}\min_{\boldsymbol{w}} \quad & -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w} \\ \text{s.t.} \quad & \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1\end{aligned}$$

运用拉格朗日乘子法，有 $\mathbf{S}_b\boldsymbol{w} = \lambda\mathbf{S}_w\boldsymbol{w}$

$\mathbf{S}_b\boldsymbol{w}$ 的方向恒为 $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$，不妨令 $\mathbf{S}_b\boldsymbol{w} = \lambda(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$

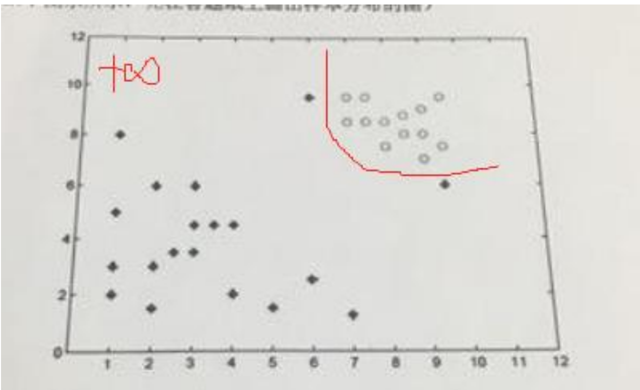
于是 $\boldsymbol{w} = \mathbf{S}_w^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$

实践中通常是进行奇异值分解 $\mathbf{S}_w = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ $\longrightarrow \beta$

然后 $\mathbf{S}_w^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

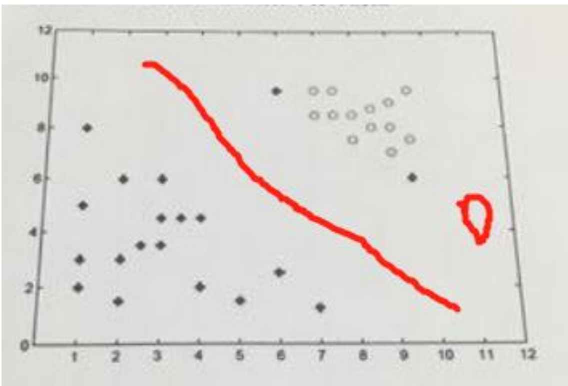
8.SVM

1) C 取无穷大分类边界



惩罚因子 C 越大，则 SVM 会更倾向把所有数据分对，往往出现较小的 margin，最终导致过拟合现象，泛化性能不好。

2) C 取无穷小时的边界



C 过于小，则惩罚力度不够，SVM 会更倾向实现最大化的 margin，而对样本分对分错不关心，不利于分类。

3) 综上来看，在测试集中，C 取无穷小的时候效果会相对更好一些。

4) 图 d

原因：没有核函数，所以是直线，然后因为 C 比较小，所以倾向于最大化 margin

5) 图 a

原因：存在核函数，所以是曲线，而且西格玛比较小，所以倾向于不复杂的分类边界。