第三章 线性模型

王星

中国人民大学统计学院

April 13, 2021

大纲

- 基本形式
- 线性回归
- Logistic回归（对数几率回归）
- 线性判别分析
- 多分类学习
- 类别不平衡问题

基本形式

- 给定d个属性描述的示例$\mathbf{x} = (x_1; \cdots ; x_d)$,线性模型试图通过建立一个属性的线性组合函数来预测
- 模型:

$$y = f(x, w) = \sum_{j=1}^{d} w_j x_j + b;$$

- 用向量表示为$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$; 其中$\mathbf{w} = (w_1; \cdots ; w_d)$.模型的学习参数是$d$个$\mathbf{w}$ 和b.
- Here the x's might be
  - Raw predictor variables (continuous or coded-categorical);
  - Transformed predictors ($x_4 = \log x_3$)
  - Basis expansions ($x_4 = x_3^2, x_5 = x_3^3, etc.$)
  - Interactions ($x_4 = x_2 x_3$.)
- 线性模型的特点：简单，易于建模，许多非线性模型都是在线性模型的基础上通过引入层级结构或高维映射而获得。
- 反映了属性在预测中的重要性，当系数估计标准差相等的时候，其大小和符号表示了对预测影响的大小和方向。

一元线性回归解法1/2(截距项$b = 0$的情况)

- Suppose that we have observations $y = (y_1, ..., y_m) \in R^m$, and we want to model these a linear function of $x = (x_1, ..., x_m) \in R^m$,
- 损失函数This value $\hat{w} \in R$ is optimal in the least squares sense:

$$\hat{w}^* = \text{argmin}_w \sum_{i=1}^{m} (y_i - f(x_i))^2 = \text{argmin}_w \sum_{i=1}^{m} (y_i - wx_i)^2$$
$$= \text{argmin}_w ||y - wx||_2^2.$$

- The univariate linear regression coefficient of y on x is

$$\hat{w} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{x^T y}{||x||_2^2}$$

We often think of the observations y as coming from the model

$$y = w * x + \epsilon.$$

where $x \in R^n$ are fixed (nonrandom) measurements, $w^* \in R$ is some true coefficient, and $\epsilon = (\epsilon_1, ..., \epsilon_n) \in R^n$ are errors with $\text{E}[\epsilon_i] = 0, \text{Var}(\epsilon_i) = \sigma^2, \text{Cov}(\epsilon_i, \epsilon_j) = 0.$

一元回归系数估计及意义Review: univariate regression2/2

Now add an intercept term to the linear model:

$$y = b^* + w^* x + \epsilon$$

Estimate $(\hat{b}, \hat{w})$ using least squares,

$$(\hat{b}, \hat{w}) = \text{argmin}_{\hat{b},\hat{w}} \sum_{i=1}^{m} (y_i - b - wx_i)^2 = \text{argmin}_{b,w} \|y - b\mathbb{1} - wx\|^2$$

giving:

$$\hat{b} = \bar{y} - \hat{w}\bar{x}, \quad \hat{w} = \frac{(x - \bar{x}\mathbb{1})^T (y - \bar{y}\mathbb{1})}{\|x - \bar{x}\mathbb{1}\|_2^2}$$

Notice that

$$\hat{w} = \frac{\text{cov}(x, y)}{\text{var}(x)} = \text{cor}(x, y) \sqrt{\frac{\text{var}(y)}{\text{var}(x)}}$$

Review: multivariate regression1/2

- Now suppose that we are considering $y \in \mathbb{R}^m$ as a function of multiple predictors $X_1, \ldots, X_d \in \mathbb{R}^m$. We collect these predictors into columns of a predictor matrix $X \in \mathbb{R}^{m \times d}$. We assume that $X_1, \ldots, X_d$ are linearly independent($d \leq m$), so that $\text{rank}(X) = d$
- The model
$$f(\mathbf{x_i}) = \mathbf{w}^T \mathbf{x_i} + b$$
where $X \in \mathbb{R}^{m \times d}$ is considered fixed, $w^* = (w_1^*, \ldots, w_d^*) \in \mathbb{R}^d$ are the true coefficients, and the errors $\epsilon = (\epsilon_1, \ldots, \epsilon_m) \in \mathbb{R}^m$ are as before (i.e., satisfying $\text{E}[\epsilon] = 0$ and $\text{Cov}(\epsilon) = \sigma^2 I$)
- For an intercept term, we can just append a column $\mathbb{1} \in \mathbb{R}^m$ of all 1s to the matrix X
- Estimate the coefficients $\hat{w} \in \mathbb{R}^d$ by least squares:
$$\hat{w} = \text{argmin}_{w \in R^d} \|y - X\hat{w}\|_2^2 = \text{argmin}_{w \in R^d} (\mathbf{y} - \mathbf{Xw})^T (\mathbf{y} - \mathbf{Xw})$$
- This gives
$$\hat{w} = (X^T X)^{-1} X^T y$$

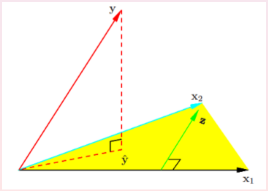
Geometry of Multivariate regression2/2

The fitted values are

$$\hat{y} = X\hat{w} = X(X^T X)^{-1} X^T y$$

This is a linear function of y, $\hat{y} = Hy$, where $H = X(X^T X)^{-1} X^T$ is sometimes called the hat matrix.
The linear regression $\hat{y} \in \mathbb{R}^m$ is exactly the projection of $y \in \mathbb{R}^m$ onto the linear subspace $\text{span}\{X_1, \ldots, X_d\} = \text{col}(X) \subseteq \mathbb{R}^m$.
projection matrix:

- The matrix H is symmetric: $H^T = H$;
- idempotent: $H^2 = H$;
- $Hx = x$ for all $x \in \text{col}(X)$, and $Hx = 0$ for all $x \perp L$.



Orthogonal complement

E.g., for any subspace $L \subseteq \mathbb{R}^m$, its orthogonal complement is $L^\perp = \{x \in \mathbb{R}^m : x \in L\} = \{x \in \mathbb{R}^m : x \perp v \text{ for any } v \in L\}$.
Fact: $P_L + P_{L^\perp} = I$, so that $P_{L^\perp} = I - P_L$.

Hence for the linear regression of y on X, the residual vector is

$$y - \hat{y} = (I - P_{\mathrm{col}(X)})y = P^\perp_{\mathrm{col}(X)} y$$

So $y - \hat{y}$ is orthogonal to any $v \in \mathrm{col}(X)$; In particular, this means that $y - \hat{y}$ is orthogonal to each of $X_1, \ldots, X_d$.
E.g., the projection map P_L onto any linear subspace $L \in \mathbb{R}^m$ is always non-expansive, that is, for any points $x, z \in \mathbb{R}^m$,

$$\|P_L x - P_L z\|_2 \leq \|x - z\|_2;$$

Hence if $y_1, y_2 \in \mathbb{R}^m$ and $\hat{y}_1, \hat{y}_2 \in \mathbb{R}^m$ are their regression fits, then

$$\|\hat{y}_1 - \hat{y}_2\| = \|P_{\mathrm{col}(X)} y_1 - P_{\mathrm{col}(X)} y_2\|_2 \leq \|y_1 - y_2\|_2;$$

Univariate regression with intercept

For univariate linear regression with an intercept term, i.e., for regressing $y \in \mathbb{R}^n$ on predictors $\mathbb{1}, x \in \mathbb{R}^n$,we can write the coefficient of x as

$$\hat{w}_1 = \frac{\langle x - \bar{x}\mathbb{1}, y\rangle}{\|x - \bar{x}\mathbb{1}\|_2^2}$$

We can alternatively view this as result of two steps:

- Regress x on $\mathbb{1}$, yielding the coefficient

$$\frac{\langle \mathbb{1}, x\rangle}{\|\mathbb{1}\|_2^2} = \frac{\langle \mathbb{1}, x\rangle}{n} = \bar{x}$$

  and the residual $z = x - \bar{x}\mathbb{1} \in \mathbb{R}^n$

- Regress y on z, yielding the coefficient

$$\hat{w}_1 = \frac{\langle z, y\rangle}{\|z\|_2^2} = \frac{\langle x - \bar{x}\mathbb{1}, y\rangle}{\|x - \bar{x}\mathbb{1}\|_2^2}$$

Multivariate regression by orthogonalization 1/3

This idea extends to multivariate linear regression of $y \in \mathbb{R}^m$ on predictors $X_1, \ldots, X_d \in R^m$. Consider the p-step procedure:

1. Let $Z_1 = X_1$;
2. For $j = 2, \ldots, d$: Regress X_j onto $Z_1, \ldots, Z_{j-1}$ to get coefficients $\hat{\gamma}_{jk} = \frac{\langle Z_k, X_j \rangle}{\|Z_k\|_2^2}$ for $k = 1, \ldots, j-1$, and residual vector
$$Z_j = X_j - \sum_{k=1}^{j-1} \hat{\gamma}_{jk} Z_k$$
3. Regress y on Z_d to get the coefficient $\hat{w}_d$.

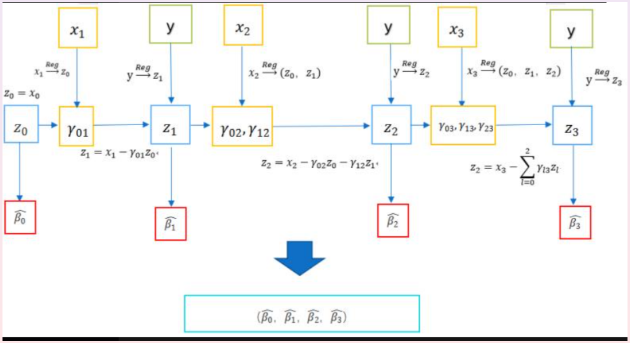
Multivariate regression by orthogonalization 2/4

The model

$$f(\mathbf{x_i}) = \beta^T \mathbf{x_i} + b$$

where $X \in \mathbb{R}^{m \times d}$ is considered fixed, $\beta^* = (\beta_1^*, \ldots, \beta_d^*) \in \mathbb{R}^d$ are the true coefficients, and the errors $\epsilon = (\epsilon_1, \ldots, \epsilon_m) \in \mathbb{R}^m$ are as before (i.e., satisfying $\mathrm{E}[\epsilon] = 0$ and $\mathrm{Cov}(\epsilon) = \sigma^2 I$)

Input: $X_1, \cdots, X_d$, Output: $\hat{\beta}_q, q = 1, \cdots, d$


x_1
y
x_2
y
x_3
y
$x_1 \xrightarrow{Reg} z_0$
$y \xrightarrow{Reg} z_1$
$x_2 \xrightarrow{Reg} (z_0, z_1)$
$y \xrightarrow{Reg} z_2$
$x_3 \xrightarrow{Reg} (z_0, z_1, z_2)$
$y \xrightarrow{Reg} z_3$
$z_0 = x_0$
z_0
γ_{01}
z_1
γ_{02}, γ_{12}
z_2
$\gamma_{03}, \gamma_{13}, \gamma_{23}$
z_3
$z_1 = x_1 - \gamma_{01} z_0$
$z_2 = x_2 - \gamma_{02} z_0 - \gamma_{12} z_1$
$z_2 = x_3 - \sum_{l=0}^{2} \gamma_{l3} z_l$
$\widehat{\beta_0}$
$\widehat{\beta_1}$
$\widehat{\beta_2}$
$\widehat{\beta_3}$
$(\widehat{\beta_0}, \widehat{\beta_1}, \widehat{\beta_2}, \widehat{\beta_3})$

Multivariate regression by orthogonalization 3/4

1. The vectors $Z_1, \ldots, Z_d \in \mathbb{R}^m$ produced by this algorithm are orthogonal.
2. For any $j = 1, \ldots, d$, the definition $Z_j = X_j - \sum_{k=1}^{j-1} \gamma_{jk} Z_k$ shows that each Z_j is a linear combination of $X_1, \ldots, X_j$, In fact, $\text{span}\{X_1, \ldots, X_j\} = \text{span}\{Z_1, \ldots, Z_j\}$.
3. the linear regression fit y on $X_1, \ldots, X_d$ is the same as the linear regression fit of y on $Z_1, \ldots, Z_d$. Call this fit $\hat{y}$.
$$y = c_1 Z_1 + \ldots + c_d Z_d;$$
for some $c_1, \ldots, c_d$
4. As $Z_1, \ldots, Z_d$ are orthogonal, the coefficients $c_1, \ldots, c_d$ are just given by univariate linear regression, so in particular we have
$$c_d = \frac{\langle Z_d, y \rangle}{\|Z_d\|_2^2} = \hat{w}_d.$$
5. For each Z_j in the expression
$$\hat{y} = c_1 Z_1 + \ldots + c_{d-1} Z_{d-1} + \hat{w}_d Z_d$$

Multivariate regression by orthogonalization 4/4

plug in the linear representation in terms of $X_1, \ldots, X_d$. Note that the variable X_d appears only through Z_d, and the coefficient of X_d is $\mathbb{1} : Z_d = X_d - \sum_{k=1}^{d-1} \hat{\gamma}_{dk} Z_k$.

Claim

the output $\hat{w}_d$ of this algorithm is exactly the coefficient of X_d in the multivariate linear regression of y on $X_1, \ldots, X_d$.

$$\hat{w}_d = \frac{\langle Z_d, y \rangle}{\|Z_d\|_2^2}, \quad \hat{w}_j = \frac{\langle Z_j, y \rangle}{\|Z_j\|_2^2}$$

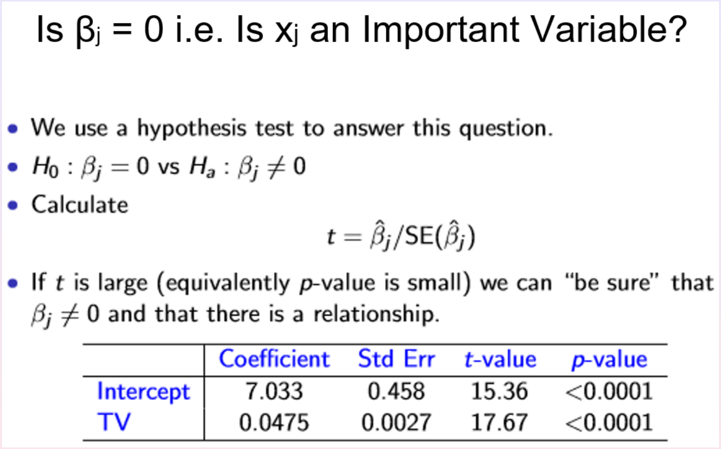
where Z_p the residual from regressing X_p onto $Z_1, \ldots, Z_{p-1}$, i.e., the residual from regressing X_d onto $X_1, \ldots, X_{d-1}$.

Orthogonal and correlated predictor variables

1 If $X_1, \ldots, X_d$ are orthogonal, then we claimed last slides that the jth multiple regression coefficient of y on $X_1, \ldots, X_d$ is equal to the univariate regression coefficient of y on X_j.

2 If $X_1, \ldots, X_d$ are correlated, Note that z_j is the residual from regressing X_j onto $X_i, i \neq j$. Remember that the regression fit of X_j onto $X_i, i \neq j$ is really just the projection of X_j onto the linear subspace $\mathrm{span}\{X_i : i \neq j\}$.

3 If X_j is highly correlated with the rest, then this fit is close to X_j, so the residual z_j is close to 0. This makes the regression coefficient $\hat{w}_j = \frac{\langle z_j, y \rangle}{\|z_j\|_2^2}$ unstable, as the denominator is very small, but the numerator can be too.



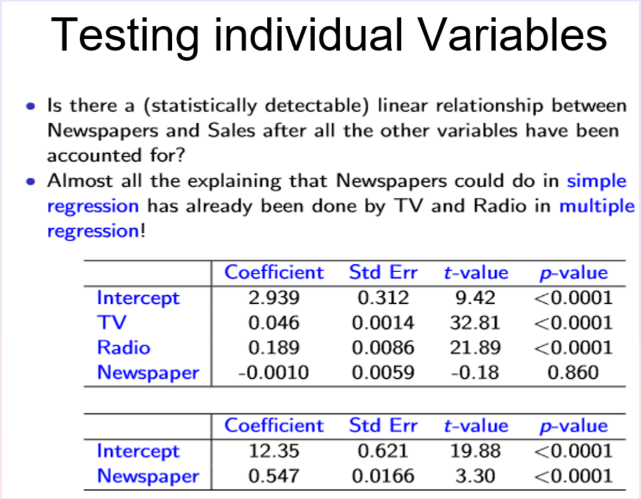
Is β$_j$ = 0 i.e. Is x$_j$ an Important Variable?

- We use a hypothesis test to answer this question.
- $H_0 : \beta_j = 0$ vs $H_a : \beta_j \neq 0$
- Calculate

$$t = \hat{\beta}_j / \mathrm{SE}(\hat{\beta}_j)$$

- If t is large (equivalently p-value is small) we can "be sure" that $\beta_j \neq 0$ and that there is a relationship.

	Coefficient	Std Err	t-value	p-value
Intercept	7.033	0.458	15.36	<0.0001
TV	0.0475	0.0027	17.67	<0.0001



Testing individual Variables

- Is there a (statistically detectable) linear relationship between Newspapers and Sales after all the other variables have been accounted for?
- Almost all the explaining that Newspapers could do in simple regression has already been done by TV and Radio in multiple regression!

	Coefficient	Std Err	t-value	p-value
Intercept	2.939	0.312	9.42	<0.0001
TV	0.046	0.0014	32.81	<0.0001
Radio	0.189	0.0086	21.89	<0.0001
Newspaper	-0.0010	0.0059	-0.18	0.860

	Coefficient	Std Err	t-value	p-value
Intercept	12.35	0.621	19.88	<0.0001
Newspaper	0.547	0.0166	3.30	<0.0001

Variance inflation

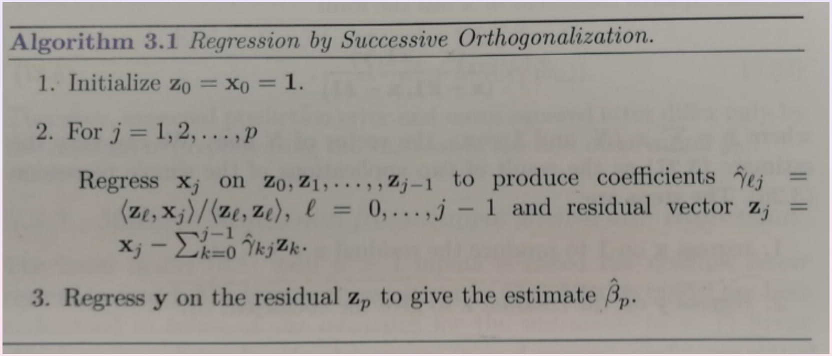
From this formula we can explicitly compute the variance of the jth multiple regression coefficient:

$$\mathrm{Var}(\hat{w}_j) = \frac{\mathrm{Var}(\langle z_j, y\rangle)}{\|z_j\|_2^4} = \frac{\|z_j\|_2^2\sigma^2}{\|z_j\|_2^4} = \frac{\sigma^2}{\|z_j\|_2^2}$$

Having correlated predictors inflates the variance of multiple regression coefficients. Remember that the Z-statistic for the jth regression coefficient is

$$Z_j = \frac{\hat{w}_j}{\sqrt{\mathrm{Var}(\hat{w}_j)}} = \frac{\hat{w}_j}{\sigma} \cdot \|z_j\|_2$$

so if X_j is highly correlated with the other predictors, its regression coefficient will likely be not significant (according to Z_j)



Algorithm 3.1 *Regression by Successive Orthogonalization.*

```
1. Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}$.

2. For $j = 1, 2, \ldots, p$

       Regress $\mathbf{x}_j$ on $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle \mathbf{z}_\ell, \mathbf{x}_j \rangle / \langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle$, $\ell = 0, \ldots, j-1$ and residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$.

3. Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ to give the estimate $\hat{\beta}_p$.
```

线性回归的变型

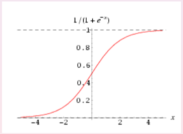
- 线性回归函数可以简写为：$y = wx + b$，这是用模型的预测值去逼近真实标记y。假设所对应的输出标记是在指数尺度上是有变化的，那么就可将输出标记的对数作为线性模型逼近的目标，即可得到“对数线性回归”（log-linear regression）：
- $\ln(y) = wx + b$
- 更一般地，考虑单调可微函数g，令：$g(y) = wx + b$这样得到的模型称为“广义线性模型”（generalized linear model），其中函数g称为“连接函数”（link function）。不同的连接函数会构成不同的线性回归模型。其中，当$g(y) = \ln(y/(1-y))$ 时，我们得到了“对数几率回归”（logit regression），也称“逻辑回归函数”。通过连接函数，可以看出该模型实际上是在用线性回归模型的预测结果去逼近真实标记的对数几率。

Logistic回归1

如果要进行的是分类任务，就需要将线性回归模型的预测值与分类任务的真实标记联系起来。

- 对于二分类任务来说，输出$y \in \{0, 1\}$, 线性回归模型产生的预测值是实值，需要将实值转换为0/1 值。最理想的方式就是给预测值加上一个单位阶跃函数，若预测值大于0 就判为正类，小于0 则判为负类，预测值为临界值0 则判别为任意。

$$y = \begin{cases} 0, & z \leq 0 \\ 0.5 & z = 0 \\ 1, & z > 0 \end{cases}$$



- 阶跃函数的缺点：不连续，无法微分，于是找到替代函数，对数几率函数(S 型函数)

$$y = \frac{1}{1 + e^{-x}}.$$

算法

- 初始化模型，输入$(x,y)=((x_1,y_1),...(x_m,y_m))\in R^{m\times d}$,$x_i\in$, $y_i\in\{0,1\}$.
- 输出过程：
    - 初始化模型参数$w\in\mathbb{R}^d, b\in\mathbb{R}$
    - 建立逻辑回归模型

$$p_1(x)=p(y=1|x)=\frac{e^{w^Tx+b}}{1+e^{w^Tx+b}};\ \ p(y=0|x)=\frac{1}{1+e^{w^Tx+b}}.$$

令$w=(w,b),\tilde{x}_i=(x_i,1)$,那么

$$p_1(x)=p(y=1|x)=\frac{e^{w^T\tilde{x}}}{1+e^{w^T\tilde{x}}};\ \ p(y=0|x)=\frac{1}{1+e^{w^T\tilde{x}}}.$$

$$\Rightarrow \ln\frac{p_1}{1-p_1}=w^T\tilde{x}\ \ \ln(1-p_1)=-\ln(1+e^{w^T\tilde{x}})$$

- 两点分布$P(y_i|x_i,w)=p_1^{y_i}(1-p_1)^{1-y_i}$;

$$\ln P(y_i|x_i,w)=y_i\ln p_1+(1-y_i)\ln(1-p_1)=y_i\left(\ln\frac{p_1}{1-p_1}\right)+\ln(1-p_1)$$

- 记$l(w) = \ln\prod_{i=1}^m P(y_i|x_i,w)$为对数似然函数；
- 负对数似然函数$-l(w) = \sum_{i=1}^m(-y_i w^T\tilde{x}_i + \ln(1+\exp w^T\tilde{x}_i))$
- 负对数似然函数为损失函数，为求最小值，$l(w)$ 对w求导数

$$\frac{\partial l(w)}{\partial w_j} = \sum_{i=1}^m \tilde{x}_{ij}(y_i - p_1(\tilde{x}_i;w));$$

$$\frac{\partial^2 l(w)}{\partial w_j \partial w_k^T} = \sum_{i=1}^m \tilde{x}_i\tilde{x}_i^T p_1(\tilde{x}_i;w)(1-p_1(\tilde{x}_i;w))$$

- 注意:以上求导过程中，$\ln(1+w^T\tilde{x}_i)$对w_j求导时，$p(x_i;w) = \frac{e^{w^T\tilde{x}}}{1+e^{w^T\tilde{x}}}$
- 特别的，

$$\frac{\partial^2 l(w)}{\partial w_0 \partial w_j^T} = \sum_{i=1}^m \tilde{x}_i p_1(\tilde{x}_i;w)(1-p_1(\tilde{x}_i;w))$$

- $$\frac{\partial^2 l(w)}{\partial w_0 \partial w_0^T} = \sum_{i=1}^m p_1(\tilde{x}_i;w)(1-p_1(\tilde{x}_i;w))$$

梯度下降法gradient descent最速下降法steepest descent，(1)参见李航附录A

- $f(x)$是$\mathbb{R}^m$上有一阶连续偏导数的函数，要求其无约束最优化问题

$$\min_{x\in\mathbb{R}^m} f(x), \quad x^* = \operatorname{argmin}_{x\in\mathbb{R}^m} f(x).$$

- 梯度下降法的基本原理：根据泰勒展开，

$$f(x) = f(x^{(k)}) + g_k^T(x - x^{(k)}).g_k = \nabla f(x^{(k)})\text{称为梯度}$$

- 选取适当的初值$x^{(0)}$,不断按照以下公式迭代，直到收敛：

$$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p_k.$$

- 其中p_k是搜索方向,取负梯度方向$p_k = -\nabla f(x^{(k)})$, λ_k 是步长，由一维搜索确定，即λ_k 使得

$$f(x^{(k)} + \lambda_k p_k) = \min_{\lambda\geq 0} f(x^{(k)} + \lambda p_k)$$

梯度下降法gradient descent最速下降法steepest descent(2)

- 输入：目标函数$f(x)$,梯度函数$g(x) = \nabla f(x)$, 计算精度ϵ.
- 输出：$f(x)$的极小点x^*.

1. 取初始值$x^{(0)}$,置$k = 0$
2. 计算$f(x^{(k)})$
3. 计算梯度$g_k = g(x^{(k)})$,当$\|g_k \leq \epsilon\|$时，停止迭代，令$x^* = x^{(k)}$;
   否则，令$p_k = -g(x^{(k)})$,求λ_k使得
   $$f(x^{(k)} + \lambda_k p_k) = \min_{\lambda \geq 0} f(x^{(k)} + \lambda p_k)$$
4. 置$x^{(k+1)} \leftarrow x^{(k)} + \lambda_k p_k$,计算$f(x^{(k+1)})$
   当$\|f(x^{k+1}) - f(x^k)\| < \epsilon$或$\|x^{k+1} - x^k\| < \epsilon$ 时，停止迭代，
   令$x^* = x^{(k=1)}$
5. 否则，置$k = k + 1$转(3)

牛顿法和拟牛顿法(1)

- $f(x)$是$\mathbb{R}^m$上有二阶连续偏导数的函数，要求其无约束最优化问题

$$\min_{x\in\mathbb{R}^m} f(x), \quad x^* = \mathrm{argmin}_{x\in\mathbb{R}^m} f(x).$$

- 根据泰勒展开，

$$f(x) = f(x^{(k)}) + g_k^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})H(x^{(k)})(x - x^{(k)}).$$

$g_k = \nabla f(x^{(k)})$是$f(x)$的梯度在点$x^{(k)}$的值，$H(x^{(k)})$ 是$f(x)$ 的海森矩阵(Hessen Matrix)

$$H(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]_{n\times n}$$

- 牛顿法利用了极小点的必要条件是$\nabla f(x) = 0$,每次迭代从点$x^{(k)}$ 开始，极小点满足$\nabla f(x^{(k+1)}) = 0$
- 由泰勒展开：

$$\nabla f(x) = g_k + H_k(x - x^{(k)}) \Rightarrow g_k + H_k(x^{(k+1)} - x^{(k)}) = 0$$

牛顿法和拟牛顿法(2)

- 因此，$x^{(k+1)} = x^{(k)} - H_k^{-1} g_k \Leftrightarrow x^{(k+1)} = x^{(k)} + p_k$
- 输入：目标函数$f(x)$,梯度函数$g(x) = \nabla f(x)$, 海森矩阵$H(x)$, 计算精度ϵ.
- 输出：$f(x)$的极小点x^*.
  1. 取初始值$x^{(0)}$,置$k = 0$;
  2. 计算$g_k = g(x^{(k)})$;
  3. 若$H_k = H(x^{(k)})$,并求p_k,
  $$H_k p_k = -g_k;$$
  4. 置$x^{(k+1)} = x^{(k)} + p_k$,
  5. 否则，置$k = k + 1$转(2)

$p_k = -H_k^{-1} g_k$.

算法

- 用凸优化理论：$w^{t+1} = w^t - \frac{\partial^2 l(w)}{\partial w \partial w^T}^{-1} \frac{\partial l(w)}{\partial w}$
- 其中关于w的一阶二阶导数分别为：

$$\frac{\partial l(w)}{\partial w_j} = \sum_{i=1}^{m} \tilde{x}_{ij}(y_i - p_1(\tilde{x}_i; w));$$

$$\frac{\partial^2 l(w)}{\partial w_j \partial w_k^T} = \sum_{i=1}^{m} \tilde{x}_i \tilde{x}_i^T p_1(\tilde{x}_i; w)(1 - p_1(\tilde{x}_i; w))$$

- 特别的，

$$\frac{\partial^2 l(w)}{\partial w_0 \partial w_j^T} = \sum_{i=1}^{m} \tilde{x}_i p_1(\tilde{x}_i; w)(1 - p_1(\tilde{x}_i; w))$$

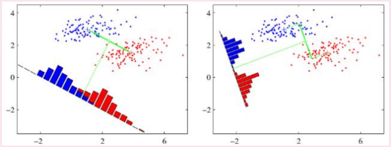
- $$\frac{\partial^2 l(w)}{\partial w_0 \partial w_0^T} = \sum_{i=1}^{m} p_1(\tilde{x}_i; w)(1 - p_1(\tilde{x}_i; w))$$

3.27思考题

- 在数据credit中，
  1. 用balance对age和limit按照矩阵求逆做回归，输出回归系数，比较和1 的异同；
  2. 用balance对rating和limit按矩阵求逆做回归，输出回归系数；
  3. 用balance对rating和limit按算法3.1做回归，输出回归系数，比较与2 和3 结果的异同，给出调整步长的估计迭代步数分析。
- 在sa心脏病数据中，用chd(1：得病，0：正常)对tobacco(吸烟量)+ldl(肥胖指数)+age(年龄)进行logistic 回归建模，预测一个人得心脏病的概率，使用梯度下降算法来得到迭代估计，返回系数估计，提供一个梯度下降的算法进行参考，在训练数据上预测数据画出决策边界。https://www.cnblogs.com/xuehen/p/5770170.html

3.4.1 线性判别分析(1)

- 线性判别分析Lineard Discriminant Analysis，LDA，是一种经典的线性学习方法，二分类问题上最早由Fisher,1936 提出，亦称“Fisher 判别分析”。
- LDA的思想：给定训练样例集，设法将样例投影到一条直线上使得同类样例的投影点尽可能接近、异类样例的投影点尽可能远离，在对新样本进行分类时，将其投影到同样的这条直线上，再根据投影点的位置来确定新样本的类别。
- 给定数据集$D = \{(x_i, y_i)_{i=1}^m, y_i = \{0, 1\}\}$,假设有两类数据，分别为红色和蓝色，如图所示，这些数据特征是二维的，希望将这些数据投影到一维的一条直线，让每一种类别数据的投影点尽可能的接近，而红色和蓝色数据中心之间的距离尽可能大。



$$J = \frac{\|w^T\mu_0 - w^T\mu_1\|_2^2}{w^T\Sigma_0 w + w^T\Sigma_1 w}$$

3.4.1 线性判别分析2

- 最优化目标函数

$$J = \frac{\|w^T\mu_0 - w^T\mu_1\|_2^2}{w^T\Sigma_0 w + w^T\Sigma_1 w} = \frac{w^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T(\Sigma_0 + \Sigma_1)w}$$

- 类内散度矩阵

$$\begin{aligned} S_w &= \Sigma_0 + \Sigma_1 \\ &= \sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T + \sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T \end{aligned}$$

- 类间散度矩阵

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T.$$

- 最优化目标函数写作：

$$J = \frac{w^T S_b w}{w^T S_w w}$$

3.4.1 线性判别分析求解3

- 不失一般性，假设$w^T S_w w = 1$,于是目标函数等价于

$$\min -w^T S_b w, \quad s.t. w^T S_w w = 1$$

- 由拉格朗日乘子法，上式等价于

$$S_b w = \lambda S_w w$$

- 其中λ是拉格朗日乘子因子，注意到$S_b w$的方向恒为$\mu_0 - \mu_1$，不妨令

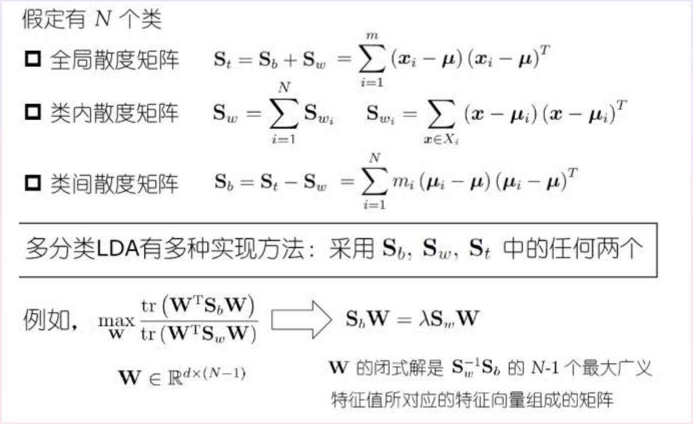
$$S_b w = \lambda(\mu_0 - \mu_1).$$

- 于是

$$w = S_w^{-1}(\mu_0 - \mu_1)$$

3.4.1 线性判别分析多分类3

- 考虑到数值解的稳定性，在实践中常常是对S_w进行奇异值分解化为对角矩阵，$S_w = U\Sigma V^T, S_w^{-1} = V\Sigma^{-1}U^T$



假定有 N 个类

- □ 全局散度矩阵 $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$
- □ 类内散度矩阵 $\mathbf{S}_w = \sum_{i=1}^{N} \mathbf{S}_{w_i} \quad \mathbf{S}_{w_i} = \sum_{\boldsymbol{x}\in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$
- □ 类间散度矩阵 $\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w = \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$

多分类LDA有多种实现方法：采用 $\mathbf{S}_b$, $\mathbf{S}_w$, $\mathbf{S}_t$ 中的任何两个

例如，$\max_{\mathbf{W}} \frac{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^T\mathbf{S}_w\mathbf{W})} \Longrightarrow \mathbf{S}_b\mathbf{W} = \lambda\mathbf{S}_w\mathbf{W}$

$\mathbf{W} \in \mathbb{R}^{d\times(N-1)}$

$\mathbf{W}$ 的闭式解是 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的 N-1 个最大广义特征值所对应的特征向量组成的矩阵

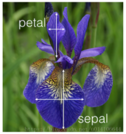
3.4.1 举例1/3

- iris案例数据，有三个类别，每个类别观察了50例，目标是寻找一个线性决策面，可以将三个类比较完整地分开

特征4个

The four features of the Iris dataset:

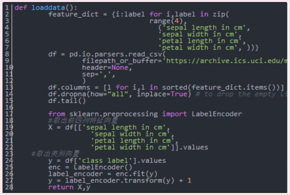
1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm



类别有3类

1. Iris-setosa (n=50)
2. Iris-versicolor (n=50)
3. Iris-virginica (n=50)

- 加载数据集分析数据loaddata() 结果返回特征向量矩阵X以及类别列向量y

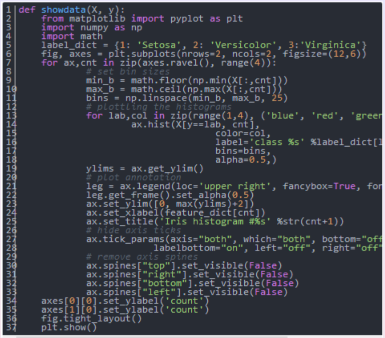


```python
def loaddata():
        feature_dict = {i:label for i,label in zip(
                        range(4),
                          ('sepal length in cm',
                          'sepal width in cm',
                          'petal length in cm',
                          'petal width in cm', ))}
        df = pd.io.parsers.read_csv(
                filepath_or_buffer='https://archive.ics.uci.edu/m
                header=None,
                sep=',',
                )
        df.columns = [l for i,l in sorted(feature_dict.items())]
        df.dropna(how="all", inplace=True) # to drop the empty li
        df.tail()

        from sklearn.preprocessing import LabelEncoder
        #取出前四列特征向量
        X = df[['sepal length in cm',
                 'sepal width in cm',
                 'petal length in cm',
                 'petal width in cm']].values
    #取出类别向量
        y = df['class label'].values
        enc = LabelEncoder()
        label_encoder = enc.fit(y)
        y = label_encoder.transform(y) + 1
        return X,y
```

3.4.1 举例2/3

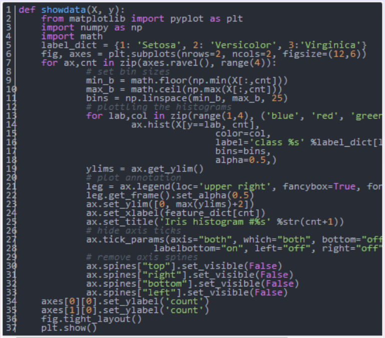
- 根据每个变量制作直方图，并以不同颜色标记不同的类别



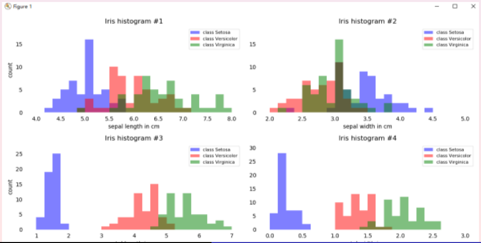
```python
def showdata(X, y):
    from matplotlib import pyplot as plt
    import numpy as np
    import math
    label_dict = {1: 'Setosa', 2: 'Versicolor', 3:'Virginica'}
    fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,6))
    for ax,cnt in zip(axes.ravel(), range(4)):
            # set bin sizes
            min_b = math.floor(np.min(X[:,cnt]))
            max_b = math.ceil(np.max(X[:,cnt]))
            bins = np.linspace(min_b, max_b, 25)
            # plottling the histograms
            for lab,col in zip(range(1,4), ('blue', 'red', 'greer
                    ax.hist(X[y==lab, cnt],
                                   color=col,
                                   label='class %s' %label_dict[l
                                   bins=bins,
                                   alpha=0.5,)
            ylims = ax.get_ylim()
            # plot annotation
            leg = ax.legend(loc='upper right', fancybox=True, for
            leg.get_frame().set_alpha(0.5)
            ax.set_ylim([0, max(ylims)+2])
            ax.set_xlabel(feature_dict[cnt])
            ax.set_title('Iris histogram #%s' %str(cnt+1))
            # hide axis ticks
            ax.tick_params(axis="both", which="both", bottom="off
                        labelbottom="on", left="off", right="off"
            # remove axis spines
            ax.spines["top"].set_visible(False)
            ax.spines["right"].set_visible(False)
            ax.spines["bottom"].set_visible(False)
            ax.spines["left"].set_visible(False)
    axes[0][0].set_ylabel('count')
    axes[1][0].set_ylabel('count')
    fig.tight_layout()
    plt.show()
```

3.4.1 举例2/3

- 根据每个变量制作直方图，并以不同颜色标记不同的类别

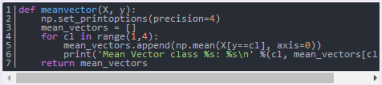


```python
def showdata(X, y):
    from matplotlib import pyplot as plt
    import numpy as np
    import math
    label_dict = {1: 'Setosa', 2: 'Versicolor', 3:'Virginica'}
    fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,6))
    for ax,cnt in zip(axes.ravel(), range(4)):
            # set bin sizes
            min_b = math.floor(np.min(X[:,cnt]))
            max_b = math.ceil(np.max(X[:,cnt]))
            bins = np.linspace(min_b, max_b, 25)
            # plottling the histograms
            for lab,col in zip(range(1,4), ('blue', 'red', 'greer
                    ax.hist(X[y==lab, cnt],
                                   color=col,
                                   label='class %s' %label_dict[l
                                   bins=bins,
                                   alpha=0.5,)
            ylims = ax.get_ylim()
            # plot annotation
            leg = ax.legend(loc='upper right', fancybox=True, for
            leg.get_frame().set_alpha(0.5)
            ax.set_ylim([0, max(ylims)+2])
            ax.set_xlabel(feature_dict[cnt])
            ax.set_title('Iris histogram #%s' %str(cnt+1))
            # hide axis ticks
            ax.tick_params(axis="both", which="both", bottom="off
                        labelbottom="on", left="off", right="off"
            # remove axis spines
            ax.spines["top"].set_visible(False)
            ax.spines["right"].set_visible(False)
            ax.spines["bottom"].set_visible(False)
            ax.spines["left"].set_visible(False)
    axes[0][0].set_ylabel('count')
    axes[1][0].set_ylabel('count')
    fig.tight_layout()
    plt.show()
```



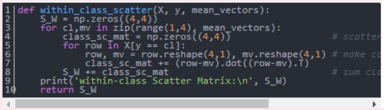
3.4.1 举例(LDA)3/3

- Step 1: Computing the d-dimensional mean vectors 计算各种类别均值μ_i;

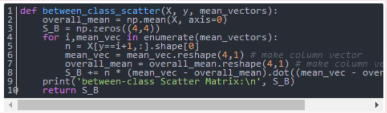


```python
def meanvector(X, y):
    np.set_printoptions(precision=4)
    mean_vectors = []
    for cl in range(1,4):
        mean_vectors.append(np.mean(X[y==cl], axis=0))
        print('Mean Vector class %s: %s\n' %(cl, mean_vectors[cl
    return mean_vectors
```

- Step 2: Computing the Scatter Matrices 计算散度矩阵,类内散度矩阵计算S_w, 类间散度矩阵S_b;
- Step 3: Solving the generalized eigenvalue problem for the matrix 求特征值分解;



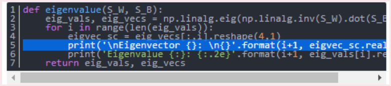
```python
def within_class_scatter(X, y, mean_vectors):
    S_W = np.zeros((4,4))
    for cl,mv in zip(range(1,4), mean_vectors):
        class_sc_mat = np.zeros((4,4))                  # scatter
        for row in X[y == cl]:
            row, mv = row.reshape(4,1), mv.reshape(4,1) # make co
            class_sc_mat += (row-mv).dot((row-mv).T)
        S_W += class_sc_mat                             # sum cla
    print('within-class Scatter Matrix:\n', S_W)
    return S_W
```



```python
def between_class_scatter(X, y, mean_vectors):
    overall_mean = np.mean(X, axis=0)
    S_B = np.zeros((4,4))
    for i,mean_vec in enumerate(mean_vectors):
        n = X[y==i+1,:].shape[0]
        mean_vec = mean_vec.reshape(4,1) # make column vector
        overall_mean = overall_mean.reshape(4,1) # make column ve
        S_B += n * (mean_vec - overall_mean).dot((mean_vec - over
    print('between-class Scatter Matrix:\n', S_B)
    return S_B
```

3.4.1 举例(LDA)

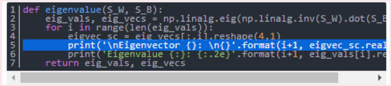
- Step 4: Selecting linear discriminants for the new feature subspace;
- 返回最大的k个特征值对应的特征向量组成的矩阵，即为W法向量子空间;
- Step 5: Transforming the samples onto the new subspace, 将样本转为到W的向量空间内，实现降维操作,并绘出样本分布图



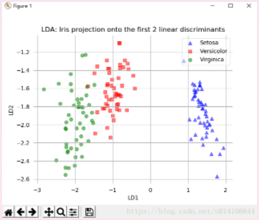
```
def eigenvalue(S_W, S_B):
    eig_vals, eig_vecs = np.linalg.eig(np.linalg.inv(S_W).dot(S_E
    for i in range(len(eig_vals)):
        eigvec_sc = eig_vecs[:,i].reshape(4,1)
        print('\nEigenvector {}: \n{}'.format(i+1, eigvec_sc.real
        print('Eigenvalue {:}: {:.2e}'.format(i+1, eig_vals[i].re
    return eig_vals, eig_vecs
```

3.4.1 举例(LDA)

- Step 4: Selecting linear discriminants for the new feature subspace;
- 返回最大的k个特征值对应的特征向量组成的矩阵，即为W法向量子空间;
- Step 5: Transforming the samples onto the new subspace, 将样本转为到W的向量空间内，实现降维操作,并绘出样本分布图



```
def eigenvalue(S_W, S_B):
    eig_vals, eig_vecs = np.linalg.eig(np.linalg.inv(S_W).dot(S_B
    for i in range(len(eig_vals)):
        eigvec_sc = eig_vecs[:,i].reshape(4,1)
        print('\nEigenvector {}: \n{}'.format(i+1, eigvec_sc.real
        print('Eigenvalue {:}: {:.2e}'.format(i+1, eig_vals[i].re
    return eig_vals, eig_vecs
```



2019.4.3作业

- 书上P69 3.5
- 用上次作业的sa心脏病数据，用chd(1:得病，0:正常)对tobacco(吸烟量)+ldl(肥胖指数)+age(年龄)进行LDA建模，假设需要判别的决策面为$y = f(x) = w^T x + w_0$，首先计算w, 估计健康人和得病病人的三个输入变量的协方差矩阵（用极大似然估计法）和每个类的均值位置μ_0和μ_1，令$\|w\| = 1$,然后根据训练数据判错率最低的要求选择w_0，返回系数估计，并且和$w = (0.61, -0.45, 0.65)^T$所给出的判别面的效果进行比较。

3.4.2 LDA学习的Bayes解决方案

对于多分类问题，假设有C个判别函数$g_c(x): c = 1, \cdots, C$, 将x判为c 类，如果满足

$$g_c(x) > g_i(x), \forall i \neq c.$$

- 令$g_c(x) = P(c|x), g_c(x) = P(x|c)P(c)$
- 将这个判别函数替换成任意一个单调函数：
$$g_c(x) = \ln P(x|c) + \ln P(c)$$
- 对于二分类问题$\{0, 1\}$, 可以定义一个决策函数
$$g(x) = g_i(x) - g_j(x)$$
- 定义：判别函数
$$\delta(x) = \begin{cases} i, & g(x) > 0 \\ j, & otherwise \end{cases}$$
- 二分类问题中通常会有的两种判决函数$g(x)$
$$g_1(x) = P(1|x) - P(0|x); \;\; g_2(x) = \ln P(1|x) - \ln P(0|x) = \ln \frac{P(x|1)}{P(x|0)} + \ln \frac{P(1)}{P(0)};$$

3.4.2 多元正态分布

- 如果$P(x|c)$服从多元正态分布

$$P(x|c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)\right]$$

这里$x = (x_1, \cdots, x_d)^T, \mu_c = (\mu_{c1}, \cdots, \mu_{cd})^T, \Sigma_{d\times d}$ 是一个协方差矩阵，$|\Sigma_c|$ 是协方差矩阵的行列式，Σ_c^{-1} 是逆。

- 决策函数可以表达为(QDA):

$$g_c(x) = -\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_c| + \ln P(c)$$

- 令$r_c^2(x) = \frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)$,

$$\delta(x) = \begin{cases} 1, & r_1^2(x) < r_0^2(x) + 2\ln\frac{P(1)}{P(0)} + \ln\frac{|\Sigma_0|}{|\Sigma_1|} \\ 0, & otherwise \end{cases}$$

$P(0)$和$p(1)$分别是0类和1类的先验概率

3.4.2 推广到多类

- 当多类$\{1, \cdots, C\}$,若$p(x|c)$是正态，Bayes Detection function
- $\delta(x) = \operatorname{argmax}(g_c(x))$

$$g_c(x) = -\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_c| + \ln P(c)$$

- 用极大似然估计表示

$$\hat{g}_c(x) = -\frac{1}{2}(x-\hat{\mu}_c)^T\hat{\Sigma}_c^{-1}(x-\hat{\mu}_c) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\hat{\Sigma}_c| + \ln P(c)$$

其中$\hat{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_{ci}$，$\hat{\Sigma}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}(x_{ci}-\hat{\mu}_c)(x_{ci}-\hat{\mu}_c)^T$

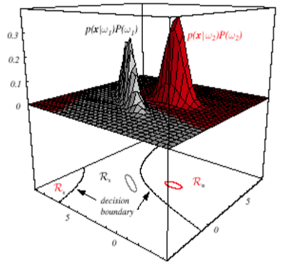
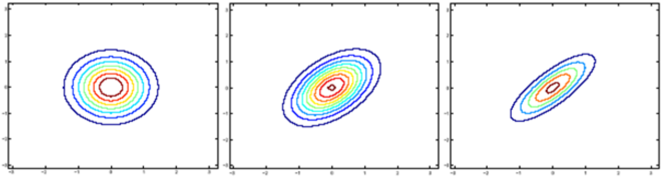


FIGURE 2.6. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision

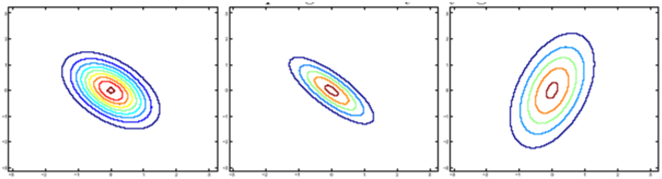
3.4.2 情形1 $\Sigma_c = \sigma^2 I$, I stands for the identity matrix(1)

- $$g_c(x) = -\frac{\|x-\mu_c\|^2}{2\sigma^2} + \ln P(c);$$
- $$\|x-\mu_c\|^2 = (x-\mu_c)^T(x-\mu_c) = x^Tx - 2\mu_c^Tx + \mu_c^T\mu_c;$$
- $g_c(x) = w_c^Tx + w_{c0}$(LDA: Linear Discriminant function线性判别函数)
  $$w_c = \frac{\mu_c}{\sigma^2}; \quad w_{c0} = -\frac{1}{2\sigma^2}\mu_c^T\mu_c + \ln P(c);$$
  w_{c0}是第c类的阈值
- 决策面（The decision surfaces）for a linear machine are pieces of hyperplanes defined by:
  $$g_i(x) = g_j(x); i \neq j \in \{1, ..., C\}$$
  $$w^T(x - x_0) = 0$$
  这里
  $$w = \mu_i - \mu_j; \quad x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2}\ln\frac{P(i)}{P(j)}(\mu_i - \mu_j)$$

3.4.2 情形1 $\Sigma_c = \sigma^2 I(2)$



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$



$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 3 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

3.4.2 情形1 $\Sigma_c = \sigma^2 I$(4)

- 用来分隔两个不同类i和j的决策面过一点x_0

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(i)}{P(j)} (\mu_i - \mu_j)$$

- 如果$P(i) = P(j)$,那么$x_0 = \frac{1}{2}(\mu_i + \mu_j)$
- 先验概率相等和不等的分界面

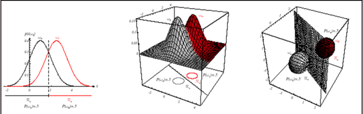


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates $\mathcal{R}_1$ from $\mathcal{R}_2$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

3.4.2 情形1 $\Sigma_c = \sigma^2 I(4)$

- 用来分隔两个不同类i和j的决策面过一点x_0

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2} \ln \frac{P(i)}{P(j)} (\mu_i - \mu_j)$$

- 如果$P(i) = P(j)$,那么$x_0 = \frac{1}{2}(\mu_i + \mu_j)$
- 先验概率相等和不等的分界面

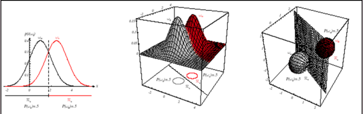
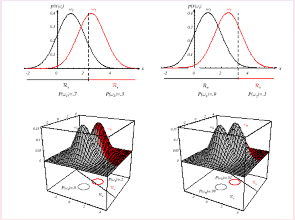
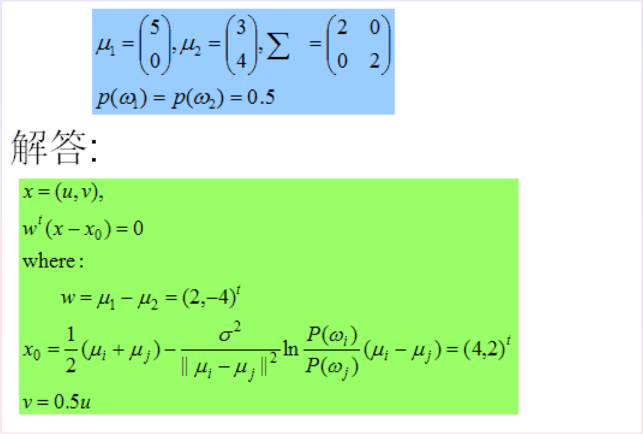


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates $\mathcal{R}_1$ from $\mathcal{R}_2$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



3.4.2 情形1 $\Sigma_c = \sigma^2 I$求二元正态分布的分界面(5)



$$\mu_1 = \begin{pmatrix} 5 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \sum = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$p(\omega_1) = p(\omega_2) = 0.5$$

解答:

$x = (u, v),$

$w^t (x - x_0) = 0$

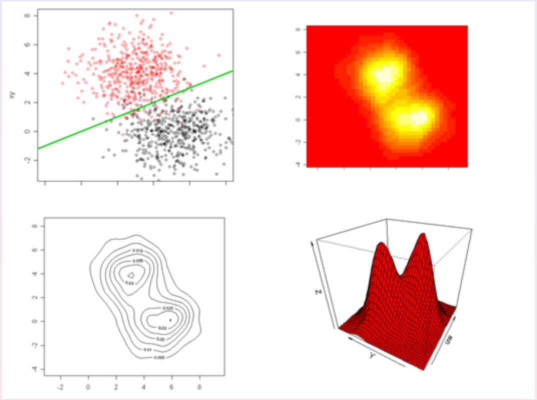
where :

$w = \mu_1 - \mu_2 = (2,-4)^t$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\| \mu_i - \mu_j \|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j) = (4,2)^t$$

$v = 0.5u$

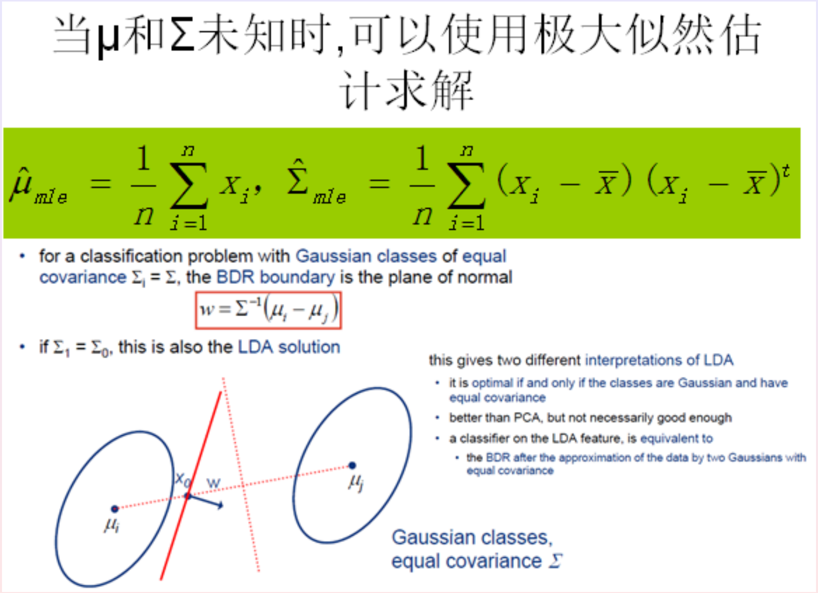
3.4.2 情形1 $\Sigma_c = \sigma^2 I$求二元正态分布的分界面(6)



3.4.2 情形2 $\Sigma_c = \Sigma$线性判别函数,协方差相等(1)

- $$g_c(x) = -\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma| + \ln P(c);$$
- $g_c(x) = w_c^T x + w_{c0}$(LDA: Linear Discriminant function线性判别函数)
$$w_c = \Sigma^{-1}\mu_c; \quad w_{c0} = -\frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \ln P(c);$$
- $w_c^T(x - x_0) = 0$这里$w = \Sigma^{-1}(\mu_i - \mu_j)$
- 用于区分第i类和第j类的分界面:
$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln(P(i)/P(j))}{(\mu_i-\mu_j)^T\Sigma^{-1}(\mu_i-\mu_j)}(\mu_i - \mu_j)$$

3.4.2 情形2 $\Sigma_c = \Sigma(2)$



当μ和Σ未知时,可以使用极大似然估计求解

$$\hat{\mu}_{mle} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \hat{\Sigma}_{mle} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^t$$

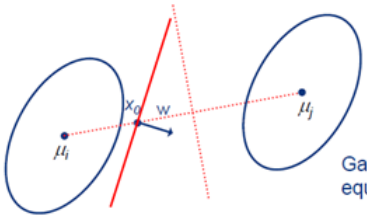
- for a classification problem with Gaussian classes of equal covariance $\Sigma_i = \Sigma$, the BDR boundary is the plane of normal

$$\boxed{w = \Sigma^{-1}(\mu_i - \mu_j)}$$

- if $\Sigma_1 = \Sigma_0$, this is also the LDA solution

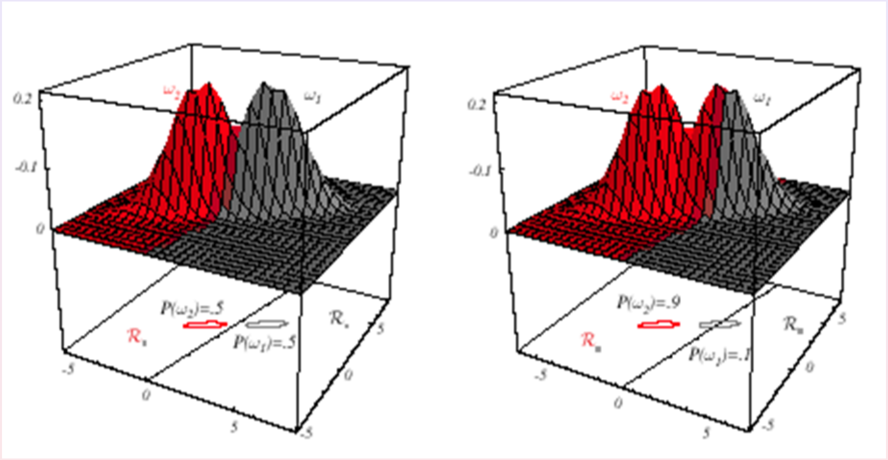
this gives two different interpretations of LDA

- it is optimal if and only if the classes are Gaussian and have equal covariance
- better than PCA, but not necessarily good enough
- a classifier on the LDA feature, is equivalent to
  - the BDR after the approximation of the data by two Gaussians with equal covariance

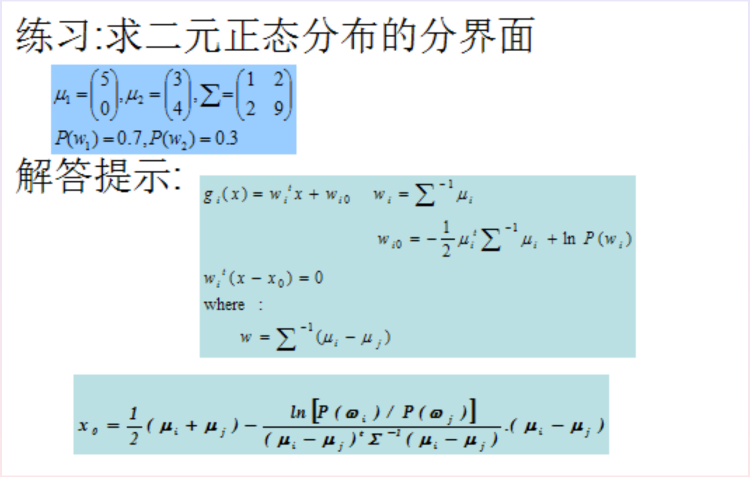


Gaussian classes, equal covariance Σ

3.4.2 情形2 $\Sigma_c = \Sigma(3)$



3.4.2 情形2 $\Sigma_c = \Sigma(4)$



练习:求二元正态分布的分界面

$$\mu_1 = \begin{pmatrix} 5 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 2 \\ 2 & 9 \end{pmatrix}$$

$$P(w_1) = 0.7, P(w_2) = 0.3$$

解答提示:

$$g_i(x) = w_i^t x + w_{i0} \quad w_i = \Sigma^{-1} \mu_i$$

$$w_{i0} = -\frac{1}{2} \mu_i^t \Sigma^{-1} \mu_i + \ln P(w_i)$$

$$w_i^t (x - x_0) = 0$$

where :

$$w = \Sigma^{-1} (\mu_i - \mu_j)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln \left[P(\omega_i) / P(\omega_j)\right]}{(\mu_i - \mu_j)^t \Sigma^{-1} (\mu_i - \mu_j)} . (\mu_i - \mu_j)$$

3.4.2 情形3 Σ_c任意(1)

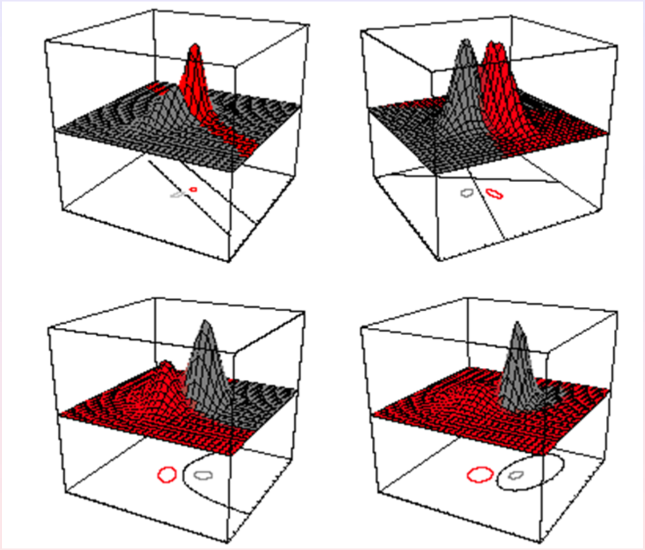
- 每一类的协方差矩阵都不同

$$g_c(x) = x^T W_c x + w_c^T x + w_{c0}$$

这里

- $W_c = -\frac{1}{2}\Sigma_c^{-1}$;
- $w_c = \Sigma_c^{-1}\mu_c$;
- $w_{c0} = -\frac{1}{2}\mu_c^T\Sigma_c^{-1}\mu_c - \frac{1}{2}\ln|\Sigma_c| + \ln P(c)$;
- 超二次曲面Hyperquadrics which are: hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, hyperhyperboloids

3.4.2 情形3 Σ_c任意(2)



3.4.2 情形3 Σ_c任意(3)

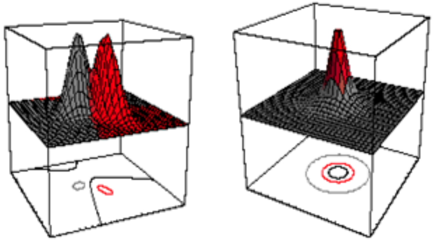
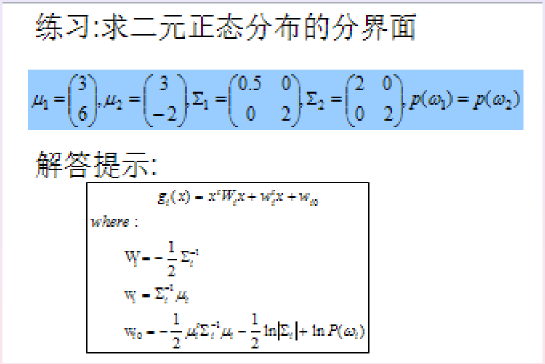


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

3.4.2 情形3 Σ_c =任意(4)


练习:求二元正态分布的分界面

$$\mu_1=\begin{pmatrix}3\\6\end{pmatrix},\mu_2=\begin{pmatrix}3\\-2\end{pmatrix},\Sigma_1=\begin{pmatrix}0.5&0\\0&2\end{pmatrix},\Sigma_2=\begin{pmatrix}2&0\\0&2\end{pmatrix},p(\omega_1)=p(\omega_2)$$

解答提示:

$$g_i(x)=x^tW_ix+w_i^tx+w_{i0}$$

where :

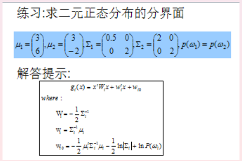
$$W_i=-\frac{1}{2}\Sigma_i^{-1}$$

$$w_i=\Sigma_i^{-1}\mu_i$$

$$w_{i0}=-\frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i-\frac{1}{2}\ln|\Sigma_i|+\ln P(\omega_i)$$

3.4.2 情形3 Σ_c =任意(4)

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


练习:求二元正态分布的分界面

$$\mu_1=\begin{pmatrix}3\\6\end{pmatrix},\mu_2=\begin{pmatrix}3\\-2\end{pmatrix},\Sigma_1=\begin{pmatrix}0.5&0\\0&2\end{pmatrix},\Sigma_2=\begin{pmatrix}2&0\\0&2\end{pmatrix},p(\omega_1)=p(\omega_2)$$

解答提示:

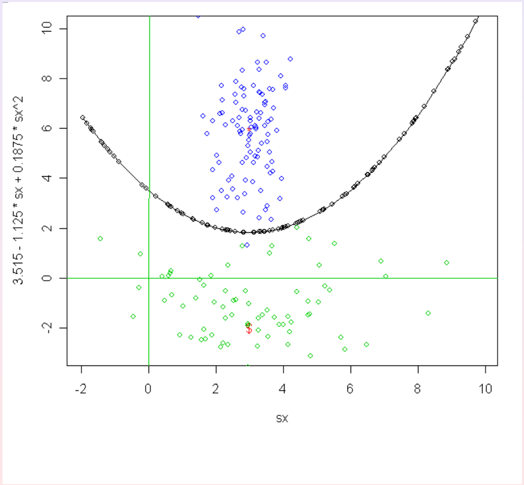
$$g_i(x)=x^tW_ix+w_i^tx+w_{i0}$$

where :

$$W_i=-\frac{1}{2}\Sigma_i^{-1}$$

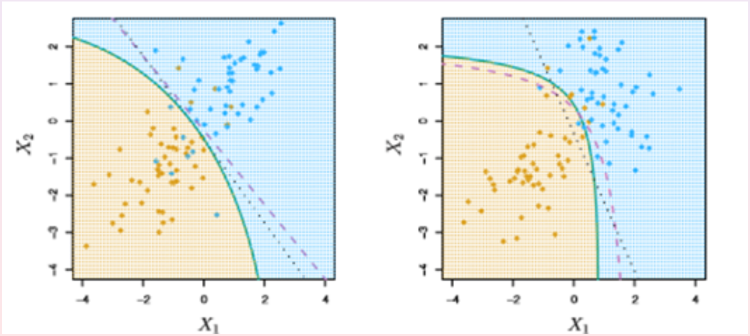
$$w_i=\Sigma_i^{-1}\mu_i$$

$$w_{i0}=-\frac{1}{2}\mu_i^t\Sigma_i^{-1}\mu_i-\frac{1}{2}\ln|\Sigma_i|+\ln P(\omega_i)$$



3.4.2 总结

- QDA will work best when the variances are very different between classes and we have enough observations to accurately estimate the variances.
- LDA will work best when the variances are similar among classes or we don't have enough data to accurately estimate the variances.



3.5 多分类学习

LDA算法既可以用来降维，又可以用来分类，但是目前来说，主要还是用于降维。在进行图像识别相关的数据分析时，LDA 是一个有力的工具。

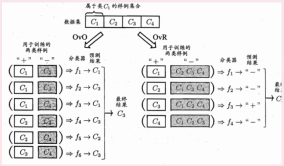
优点:

- 在降维过程中可以使用类别的先验知识经验;
- LDA在样本分类信息依赖均值而不是方差。

缺点:

- LDA不适合对非高斯分布样本进行降维。
- LDA降维最多降到类别数$k-1$的维数，如果降维的维度大于$k-1$，则不能使用LDA。当然目前有一些LDA 的进化版算法可以绕过这个问题。
- LDA在样本分类信息依赖方差而非均值时，降维效果不好。
- LDA可能过度拟合数据。

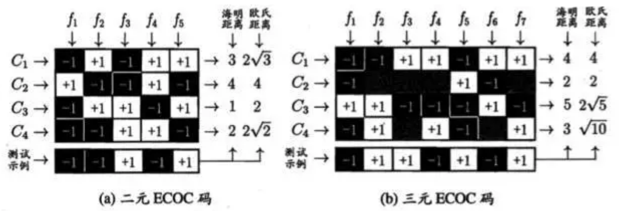
- 现实中常遇到多分类学习任务，有些二分类学习方法可直接推广到多分类，但在更多情况下，我们是基于一些基本策略，根据二分类学习器来解决多分类问题。假设有N类别$C_1, C_2, \cdots, C_N$，多分类学习的基本思路是“拆解法”，将多分类任务拆分为若干个二分类任务求解。具体来说，先对问题进行拆分，然后为拆出的每个二分类任务训练一个分类器。在测试的时候，对这些分类器的预测结果进行集成以获得最终的多分类结果。因此，如何对多分类任务进行拆分是关键。
    - 拆分策略：一对一（One vs One，简称OvO）;OvO将N个类别两两配对，从而产生N(N-1)/2个二分类任务，例如OvO将为区分类别Ci，Cj训练一个分类器，该分类器把D 中的Ci类样例作为正例，Cj类样例作为反例。测试阶段，新样本将同时提交给所有分类器，于是我们将得到N(N-1)/2个分类结果，最终结果可以通过投票产生
    - 一对其余（One vs Rest,简称OvR）;OvR则是每次将一个类的样例作为正例，所有其他类的样例作为反例来训练N个分类器，在测试时若仅有一个分类器预测为正类，则对应的类别标记作为最终分类结果。


属于类C_1的样例集合
数据集
C_1 C_2 C_3 C_4
OvO
OvR
用于训练的两类样例
“+” “−”
分类器
预测结果
$(C_1 \; C_2) \Rightarrow f_1 \rightarrow C_1$
$(C_1 \; C_3) \Rightarrow f_2 \rightarrow C_3$
$(C_1 \; C_4) \Rightarrow f_3 \rightarrow C_1$
$(C_2 \; C_3) \Rightarrow f_4 \rightarrow C_3$
$(C_2 \; C_4) \Rightarrow f_5 \rightarrow C_2$
$(C_3 \; C_4) \Rightarrow f_6 \rightarrow C_3$
最终结果
C_3
$(C_1 \; C_2 C_3 C_4) \Rightarrow f_1 \rightarrow$ “−”
$(C_2 \; C_1 C_3 C_4) \Rightarrow f_2 \rightarrow$ “−”
$(C_3 \; C_1 C_2 C_4) \Rightarrow f_3 \rightarrow$ “+”
$(C_4 \; C_1 C_2 C_3) \Rightarrow f_4 \rightarrow$ “−”

MvM策略

- MvM每次将若干个类作为正类，若干个其它类作为反类。OvO和OvR 可以看成是它的特例。MvM 的正、反类构造必须有特殊的设计，不能随意选取。
- 最常用的MvM技术：纠错输出码（ECOC）。ECOC是将编码的思想引入类别拆分，并尽可能在解码的过程中具有容错性。ECOC工作过程主要分为两步：
  （1）编码：对N个类别做M次划分，每次划分将一部分类别划为正类，一部分划为反类，从而形成一个二分类训练集，这样一共产生M个训练集，可训练出M 个训练器$f_m, m = 1, ..., M$。
  （2）解码：M个分类器分别对测试样本进行预测，这些预测标记组成一个编码。将这个与此编码与每个类别各自的编码进行比较，返回其中距离最小的类别为最终预测结果。

ECOC编码



f_1 f_2 f_3 f_4 f_5 海明距离 欧氏距离
C_1 → −1 +1 −1 +1 +1 → 3 $2\sqrt{3}$
C_2 → +1 −1 −1 +1 −1 → 4 4
C_3 → −1 +1 +1 −1 +1 → 1 2
C_4 → −1 −1 +1 +1 −1 → 2 $2\sqrt{2}$
测试示例 → −1 −1 +1 −1 +1
(a) 二元 ECOC 码
f_1 f_2 f_3 f_4 f_5 f_6 f_7 海明距离 欧氏距离
C_1 → −1 −1 +1 +1 −1 +1 +1 → 4 4
C_2 → −1 +1 −1 → 2 2
C_3 → +1 +1 −1 −1 −1 +1 −1 → 5 $2\sqrt{5}$
C_4 → −1 +1 +1 −1 +1 → 3 $\sqrt{10}$
测试示例 → −1 +1 +1 −1 +1 −1 +1
(b) 三元 ECOC 码

图 3.5 ECOC 编码示意图. “+1”、“−1”分别表示学习器 f_i 将该类样本作为正、反例; 三元码中 “0” 表示 f_i 不使用该类样本

- 二元:$N = 4, M = 5$;三元$N = 4, M = 7$.
- 海明距离：统计所有分类器f_m 对测试样本的分类结果和C_i类不一致的数目叫做第i类的海明距离，不一致计数加1，否则不加，结果为：0+1+1+1+0 = 3
- 欧式距离：每个分类器对测试样本的分类结果减去C_i 类的分类划分，差值的平方和的开方，结果为：根号下$\sqrt{((-1-(+1))^2+2^2+2^2+2^2+0)} = \sqrt{12} = 2\sqrt{3}$.

类别不平衡问题

- 类别不平衡性问题：分类任务中不同类别的训练样例数差别很大。
- 从线性分类器的角度讨论，使用$y = w^T x + b$对新样本进行分类时，用预测的y 与一个阈值进行比较，当$y > 0.5$ 即判别为正例，否则判别为负例。这里的y实际表达了正例的可能性（1-y是反例的可能性），0.5表明分类器认为正反例可能性相同,即

$$\delta(x) = \begin{cases} +, & \frac{y}{1-y} > 1 \\ -, & otherwise \end{cases}$$

- 如果训练集中正反例数目相差悬殊，令$m+$表示正例数目，$m-$表示反例数目，则观测几率就代表了真实几率，只要分类器的预测几率高于观测几率就判定为正例，即

$$\delta(x) = \begin{cases} +, & \frac{y}{1-y} > \frac{m+}{m-} \\ -, & otherwise \end{cases}$$

- 欠采样
  - 对训练集里的反例样本进行”欠采样”，即去除一些反例使得正反例数目接近，再进行学习。由于丢弃了很多反例，会使得训练集远小于初始训练集，所以有可能导致欠拟合;
  - 代表算法：EasyEnsemble;
  - 利用集成学习机制，每次从大多数类中抽取和少数类数目差不多的重新组合，总共构成n个新的训练集，基于每个训练集训练出一个AdaBoost 分类器(带阈值),最后结合之前训练分类器结果加权求和减去阈值确定最终分类类别.
- 过采样
  - 增加一些正例使得正反例数目接近，然后再学习,需要注意的是不能只是对初始正例样本重复采样，否则将导致严重的过拟合。
  - 代表算法：SMOTE
  - 合成新的少数样本的策略是，对每个少类a样本，从最近邻中随机选一个样本b，在a、b之间连线上随机选一点作为合成新样本。
  - 基于算法的改进：SMOTE可能导致初始样本分布有的部分更加稠密，有的部分更加稀疏，而且使得正反例的边界模糊。所以有学者提出Borderline-SMOTE 算法，将少数类样本根据距离多数类样本的距离分为noise,safe,danger三类样本集，只对danger中的样本集合使用SMOTE 算法。

2019.4.10作业

- 从正态分布$C_1 = N(\binom{-1.5}{-1}, \Sigma_1)$和$C_2 = N(\binom{2}{1}, \Sigma_2)$,
  $\Sigma_1 = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$和$\Sigma_2 = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$进行如下比较分析：
  1. 令$\rho = 0$,各类产生随机数100个，每类选择20个作为训练数据，另外80 个作为测试数据，比较LDA 和QDA在测试数据上的分类错误率，分析这两个模型哪个模型更适用？
  2. 令$\rho = -0.5$,，重新进行实验，实验方法和1类似，比较LDA和QDA 在测试数据上的分类错误率，分析这两个模型哪个模型更适用？
  3. 从$t(2)$分布中产生X_1, X_2生成50个观测作为C_3 类，此时需要将C_3 和C_2分开LDA和QDA哪一种比较理想？以上实验中的$P(C_i), i = 1, 2, 3$ 请用每一组参与训练的数据量来估计。
- 用上次作业的sa心脏病数据，用chd(1:得病，0:正常)对tobacco(吸烟量)+ldl(肥胖指数)+age(年龄)进行BayesLDA和QDA建模，和上次作业中的FisherLDA进行比较，实验中的$P(C_i), i = 1, 2, 3$ 请用每一组参与训练的数据量来估计。
- 请对鸢尾花数据参考如下代码http://sebastianraschka.com/Articles/2014_python_lda.html 进行三分类LDA 建模，请与OVO 方式的LDA进行比较，观察两种方法的效果有何不同？