第四章 分类线性模型

王星

中国人民大学统计学院

December 2, 2020

大纲

- 分类器，判别函数和决策面
- 基本形式
- 线性回归(review)*
- 二次判别
- Fisher 判别
- 线性判别分析
- 稀疏判别分析
- Bayes-Logistic回归（对数几率回归）*
    - Laplace近似
    - 模型比较
    - 预测分布
- 多分类学习
- 类别不平衡问题

分类的基本问题

1.回归问题有很好的数学性质和计算性质。分类的基本问题是将输入变量分到$\mathcal{C}_k$个类别中。最常见的情况是不同的类别之间互不相交。每个数据点分到唯一的类别中，输入空间分为决策区域，边界称为决策边界或决策面。本章考虑的线性分类模型,指的是决策面是线性模型。如果数据集可以被精准地分类，那么这个数据集称为线性可分的。

2.回归变量的目标变量是一个实数向量，在分类中，二分类就是一个目标变量$t \in \{0, 1\}$. 其中$t = 1$表示类别$\mathcal{C}_1$ 而$t = 0$表示类别$\mathcal{C}_2$,这样除了关心每个数据的类别$\mathcal{C}_1, \mathcal{C}_2$, 我们还关心属于某个类别的概率。

3.对于$K > 2$的情形，比较方便的表示方法是"1-of-K"的编码方式。$t = c(0/1, 0/1, \cdots, 0/1, 0/1)^T$,我们也关心每个类别的概率。

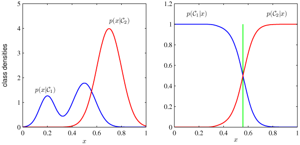
分类问题–推断和决策

有3种构造判别函数的方法

- 分布依赖型（得分依赖型）：分类问题可以分为两个阶段：推断和决策，在推断的阶段训练分布，在决策的阶段结合损失函数给出最优的分类。通常又可以分为两种方法：
  - 概率生成法（generative）
    - 用一个生成模型去推断$p(x|C_k)$;
    - 将先验分布$p(C_k)$和$p(x|C_k)$ 联合得到$p(C_k|x)$;
  - 判别模型法(discriminant):直接推断$p(C_k|x)$;
- 非概率依赖型（非得分依赖型）:直接学习一个判别函数$g(x)$.
  - 直接解一个有分类变量的映射函数；
  - 而分类问题而言，相当于一个$\{+1,-1\}$的函数

Decision theory–inference and decision

- 分布依赖型生成式模型:
  - pros: access to $p(x) \rightarrow$ easy detection of outliers I i.e., low-confidence predictions I;
  - cons: estimating the joint probability $p(x, C_k)$ can be computational and data demanding.
- 分布依赖型判别模型:
  - pros: less demanding than the generative approach;



- 判别函数:
  - pros: a single learning problem (vs inference + decision);
  - cons: no access to $p(C_k|x)$ which can have many advantages in practice for (e.g.) rejection and model combination – see page 45.

决策论-推断与决策

- The decision problem:
  - given x, predict t according to a probabilistic model $p(x, t)$
- binary classification:$t \in \{0, 1\} \leftrightarrow \{C_1, C_2\}$;
- Important quantity: $p(C_k|x)$.

$$
\begin{aligned}
p(C_k|x) = \frac{p(x, C_k)}{p(x)} &= \frac{p(x, C_k)}{\sum_{k=1}^{2} p(x, C_k)} \\
&\rightarrow \text{getting } p(x, C_i) \text{ is the (central!) inference problem} \\
&= \frac{p(x|C_k)p(C_k)}{p(x)} \propto \text{likelihood} \times \text{prior}
\end{aligned}
$$

- Intuition: choose k that maximizes $p(C_k|x)$.

决策论-二分类

- 决策域: $\mathcal{R}_i = \{x : pred(x) = C_i\}$.
- 判错率:

$$\begin{aligned} p(misclassification) &= p(x \in \mathcal{R}_1, C_2) + p(x \in \mathcal{R}_2, C_1) \\ &= \int_{\mathcal{R}_1} p(x, C_2)dx + \int_{\mathcal{R}_2} p(x, C_1)dx \end{aligned}$$

- $\Leftrightarrow p(C_1|x) > p(C_2|x)$.
- 对于k类而言: 极小化$\sum_j = \int_{\mathcal{R}_j} (\sum_{k \neq j} p(x, C_k))dx$
  $\Leftrightarrow pred(x) = \text{argmax}_k p(C_k|x)$.

分类问题–推断和决策

- 假设$\{C_i\}, i = 1, ..., K$个不同的类，如果$g_i(x) > g_j(x), i \neq j$, $g_i(x), i = 1, ..., K$ 称为判别函数;
- 例如：$g_i(x) = -R(\delta_i|x)$(这对应于后验风险最小准则)，那么根据错误最小化原理，$g_i(x) = p(C_i|x)$, 于是

$$g_i(x) = p(x|C_i)p(C_i).$$

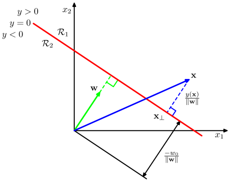
- 判别函数可以替换为任意一个单调增函数$f(g(.))$,

$$g_i(x) = \ln p(x|C_i) + \ln p(C_i).$$

分类问题中的三个空间:Three different spaces that are easy to confuse

- 权值空间M**Weight-space:**
  - Each axis corresponds to a weight;
  - A point is a weight vector;
  - Dimensionality:=♯weights +1 extra dimension for the loss.
- 数据空间D**Data-space:**
  - Each axis corresponds to an input value;
  - A point is a data vector;
  - A decision surface is a plane;
  - Dimensionality=dimensionality of a data vector;
- 样例空间N **Case-space:**
  - Each axis corresponds to a training case;
  - A point assigns a scalar value to every training case.
  - So it can represent the 1-D targets or it can represent the value of one input component over all the training data.
  - Dimensionality =♯training cases.

判别函数



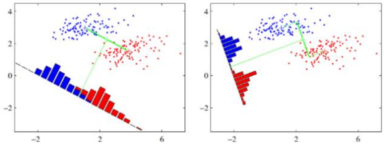
- The planar decision surface in data-space for the simple linear discriminant function:$w^T x + w_0$
- 线性判别函数$y(x) = w^T x + w_0$; 考虑两个决策面上的点x_A, x_B, 应该有$w^T(x_A - x_B) = 0$.权值向量与决策面上的任意向量都正交。
- 考虑$y(x) = 0, x \epsilon$决策面上，于是从原点出发到决策面的垂直距离是

$$\frac{w^T x}{\|w\|} = -\frac{w_0}{\|w\|}.$$

- 任意一点和它在决策面的投影$x_\perp$有$x = x_\perp + r\frac{w}{\|w\|}$， $r = \frac{y(x)}{\|w\|}$.

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

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

- 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：

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

LDA最优化目标，经常称为“广义瑞丽商”（generalized Rayleigh Quotient）

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

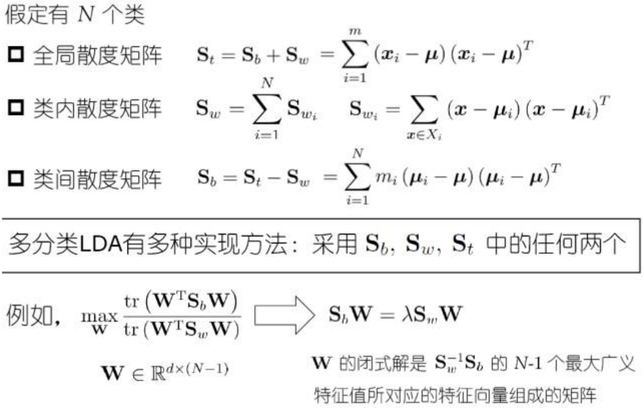
$$S_b w = \lambda(\mu_0 - \mu_1).$$

- 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

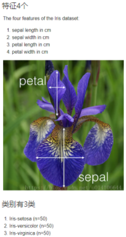
例如，$\max_{\mathbf{W}} \frac{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_b\mathbf{W}\right)}{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_w\mathbf{W}\right)} \Longrightarrow \mathbf{S}_b\mathbf{W} = \lambda\mathbf{S}_w\mathbf{W}$

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

$\mathbf{W}$ 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

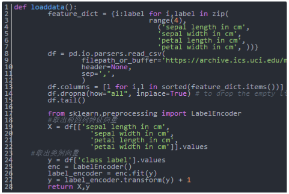
The four features of the Iris dataset:

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

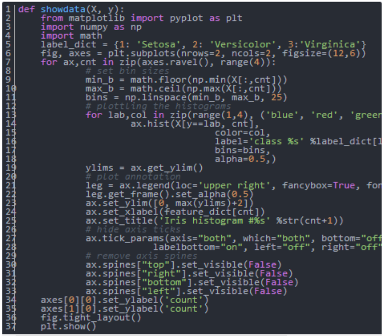


```
def loaddata():
        feature_dict = {i:label for i,label in zip(
                        range(4),
                          ('sepal length in cm',
                          'sepal width in cm',
                          'petal length in cm',
                          'petal width in cm', ))}
        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

        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

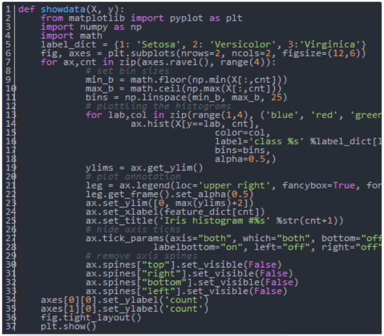
- 根据每个变量制作直方图，并以不同颜色标记不同的类别



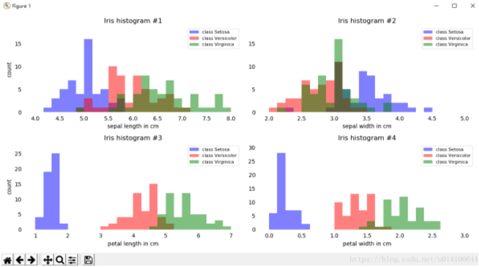
```
def showdata(X, y):
    from matplotlib import pyplot as plt
    import numpy as np
    import math
    label_dict = {1: 'Setosa', 2: 'Versicolor', 3:'Virginica'}
    fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,6))
    for ax,cnt in zip(axes.ravel(), range(4)):
            # set bin sizes
            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

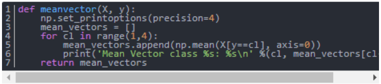


```
def showdata(X, y):
    from matplotlib import pyplot as plt
    import numpy as np
    import math
    label_dict = {1: 'Setosa', 2: 'Versicolor', 3:'Virginica'}
    fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,6))
    for ax,cnt in zip(axes.ravel(), range(4)):
            # set bin sizes
            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()
```



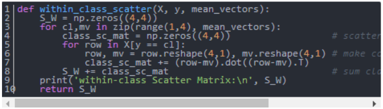
3.4.1 举例(LDA)3/3

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

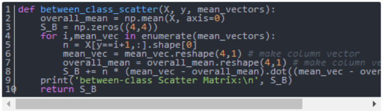


```
def meanvector(X, y):
    np.set_printoptions(precision=4)
    mean_vectors = []
    for cl in range(1,4):
        mean_vectors.append(np.mean(X[y==cl], axis=0))
        print('Mean Vector class %s: %s\n' %(cl, mean_vectors[cl
    return mean_vectors
```

- 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;



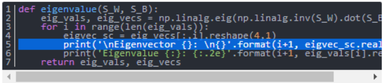
```
def within_class_scatter(X, y, mean_vectors):
    S_W = np.zeros((4,4))
    for cl,mv in zip(range(1,4), mean_vectors):
        class_sc_mat = np.zeros((4,4))                  # scatter
        for row in X[y == cl]:
            row, mv = row.reshape(4,1), mv.reshape(4,1) # make co
            class_sc_mat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



```
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)

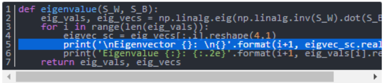
- Step 4: Selecting linear discriminants for the new feature subspace；
- 返回最大的k个特征值对应的特征向量组成的矩阵，即为W法向量子空间；
- Step 5: Transforming the samples onto the new subspace, 将样本转为到W 的向量空间内，实现降维操作,并绘出样本分布图



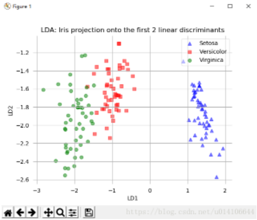
```python
def eigenvalue(S_W, S_B):
    eig_vals, eig_vecs = np.linalg.eig(np.linalg.inv(S_W).dot(S_
    for i in range(len(eig_vals)):
        eigvec_sc = eig_vecs[:,i].reshape(4,1)
        print('\nEigenvector {}: \n{}'.format(i+1, eigvec_sc.real
        print('Eigenvalue {:}: {:.2e}'.format(i+1, eig_vals[i].re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



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类决策函数$g(x): g_1(x) = P(1|x) - P(0|x)$;
  $g_2(x) = \ln P(1|x) - \ln P(0|x) = \ln \frac{P(x|1)}{P(x|0)} + \ln \frac{P(1)}{P(0)}$;

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
- $\delta(x) = \text{argmax}(g_c(x))$

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

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

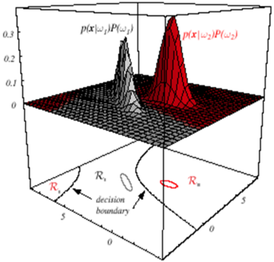
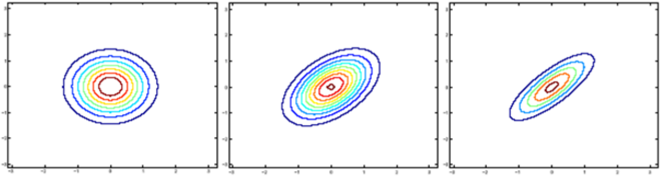


FIGURE 2.6. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region $\mathcal{R}_2$ is not simply connected. The ellipses mark where the density is $1/e$ times that at the peak of the distribution. From: Richard O. Duda, Peter E. Hart, and David G.

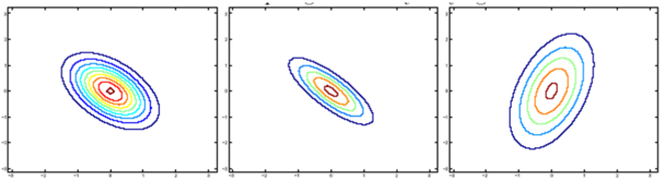
3.4.2 情形1 $\Sigma_c = \sigma^2 I$, I stands for the identity matrix(1)

- $$g_c(x) = -\frac{\|x-\mu_c\|^2}{2\sigma^2} + \ln P(c);$$
- $$\|x-\mu_c\|^2 = (x-\mu_c)^T(x-\mu_c) = x^Tx - 2\mu_c^Tx + \mu_c^T\mu_c;$$
- $g_c(x) = w_c^Tx + w_{c0}$(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 $\Sigma_c = \sigma^2 I$, I stands for the identity matrix(2)



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



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

3.4.2 情形1 $\Sigma_c = \sigma^2 I$(3)

- 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

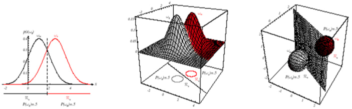


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, 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 $\Sigma_c = \sigma^2 I$(3)

- 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

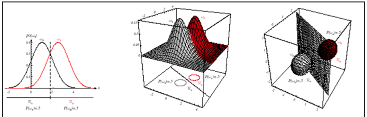
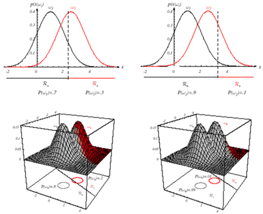
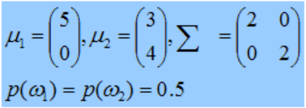


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, 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.



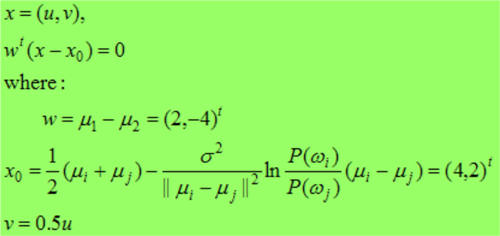
3.4.2 情形1 $\Sigma_c = \sigma^2 I$求二元正态分布的分界面(4)



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

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

解答:



$x = (u, v),$

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

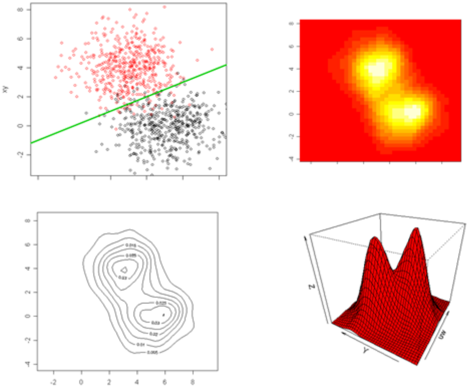
where :

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

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

$v = 0.5u$

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 $\Sigma_c = \Sigma(2)$

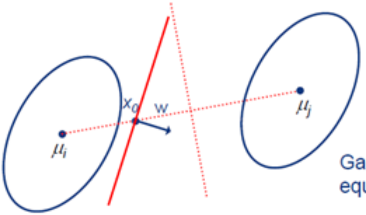
当μ和Σ未知时,可以使用极大似然估计求解

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

- 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

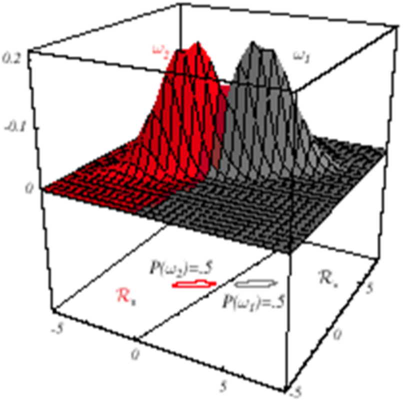


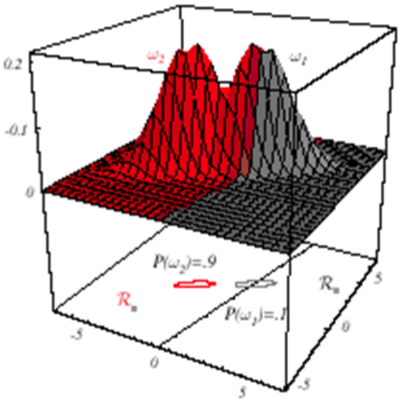
Gaussian classes, equal covariance Σ

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

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

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

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

解答提示:

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

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

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

where :

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

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

3.4.2 情形3 Σ_c任意(1)

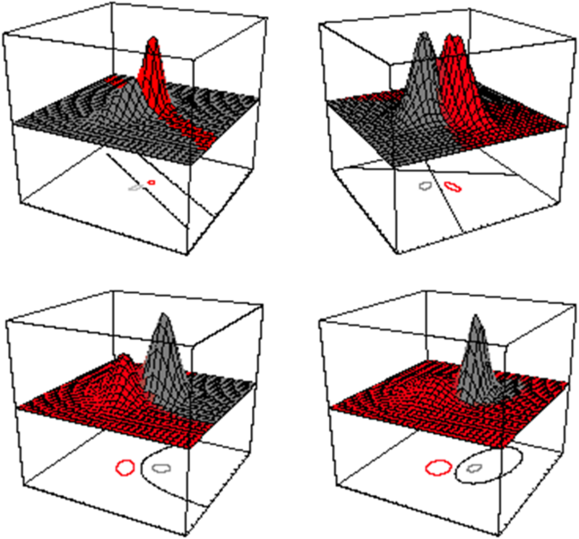
- 每一类的协方差矩阵都不同

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

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)

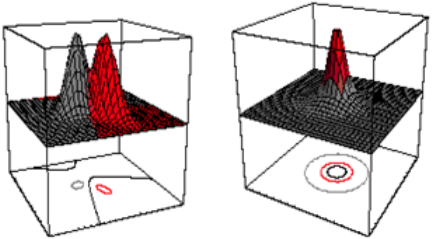
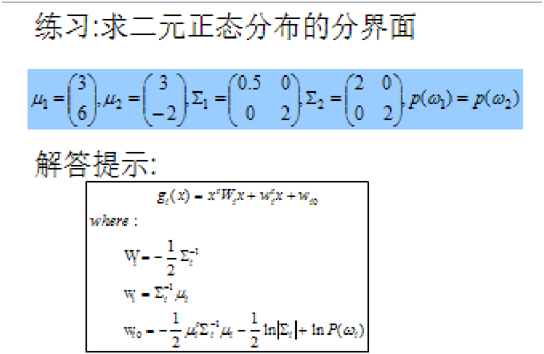


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability 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 $\Sigma_c =$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 :

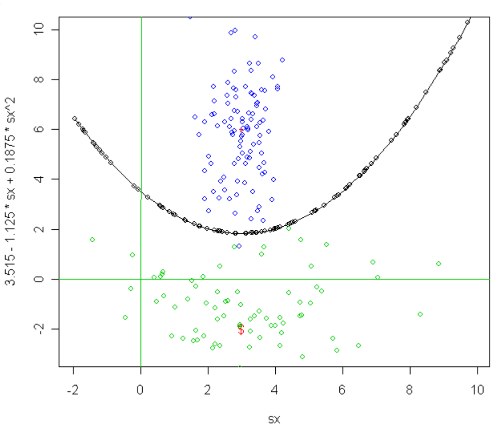
$$W_i=-\frac{1}{2}\Sigma_i^{-1}$$

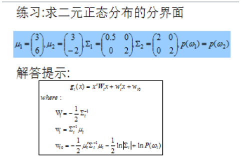
$$w_i=\Sigma_i^{-1}\mu_i$$

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

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:

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

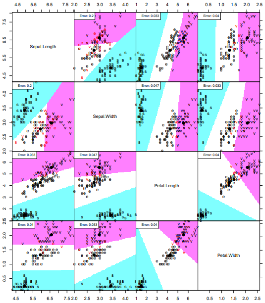
where :

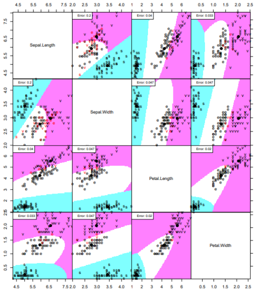
$$W_i=-\frac{1}{2}\Sigma_i^{-1}$$

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

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

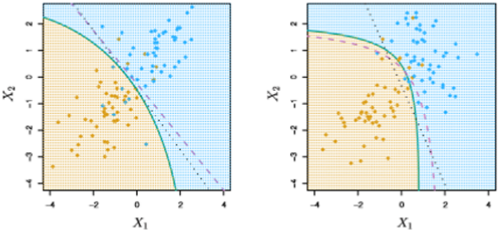
LDA at example IRIS data





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.



Linear discriminant analysis (LDA)

- A linear classifier predicts the class label of a newly given X by $\mathrm{sgn}(\eta^\top X - c)$
- LDA: Use training data to determine (η, c)
- Once $\hat{\eta} \in \mathbb{R}^p$ is obtained, it is conventional to choose $\hat{c} \in \mathbb{R}$ by minimizing the training error
- It boils down to deciding η from training data

LDA under multivariate normal model

W.l.o.g., we assume data have been centered:

$$X|y \sim \begin{cases} \mathcal{N}_p(\mu,\ \Sigma), & \text{if } y = +1, \\ \mathcal{N}_p(-\mu,\ \Sigma), & \text{if } y = -1. \end{cases}$$

- Training data: *iid* samples from both classes
- Goal: A linear classifier which predicts label of X by $\mathrm{sgn}(\hat{\eta}^\top X)$, where $\hat{\eta}$ *is from training data*
- Bayes classifier $\implies \eta \propto \Sigma^{-1}\mu$
- **Key**: Estimate $\Sigma^{-1}\mu$ from training data

Key problem

$$X_i \sim \mathcal{N}_p\big(y_i \cdot \mu,\ \Sigma\big), \quad y_i \in \{\pm 1\}, \quad i = 1, 2, ..., n$$

- Use training data to estimate $\eta = \Sigma^{-1}(\mu_1 - \mu_2)$
- Low-dimensionality:

$$\hat{\eta} = S^{-1}(\bar{X}_1 - \bar{X}_2) \qquad \text{(i.e., Fisher's LDA)}$$

Issues in high-dimensionality:

- S^{-1} is a poor estimator of $\Omega \equiv \Sigma^{-1}$
  (*precision matrix estimation*, Chapter 4)
- The vector $\bar{X}_1 - \bar{X}_2$ contains too much noise
  (*feature selection*, Chapter 2&3)

When the dimension is large,fully specifying the QDA decision boundary requires d+d(d-1)parameters,and fully specifying the LDA decision boundary requires d+d(d-1)/2parameters.Such a large number of free parameters might induce a large variance. To further regularize the model,two popular methods are diagonal quadratic discriminant analysis(DQDA) and diagonal linear discriminant analysis(DLDA).The only difference between DQDA and DLDA with QDA and LDA is that after calculating $\hat{\Sigma}_0$and $\hat{\Sigma}_1$,set all the off-diagonal elements to be zero.This is also called the Independence Rule(IR).

Shrinkage of S towards the diagonal

- Independence rule or Naive Bayes classifier

$$\hat{\eta} = D^{-1}(\bar{X}_1 - \bar{X}_2), \quad \text{where} \;\; D = \text{diag}(S)$$

- Regularized LDA *(Guo, Hastie and Tibshirani, 2006)*

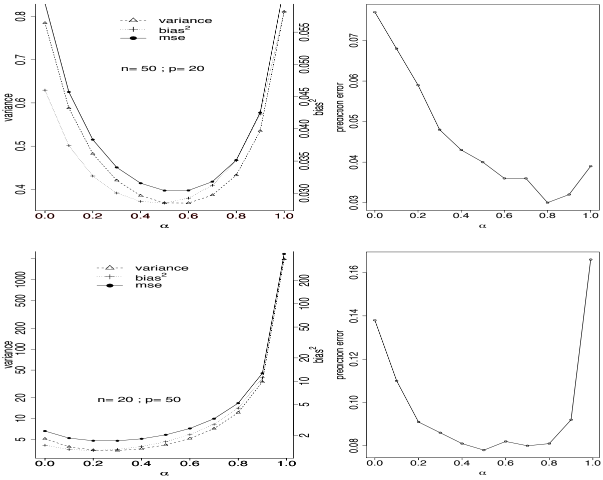
$$\hat{\eta} = [\alpha S + (1-\alpha)D]^{-1}(\bar{X}_1 - \bar{X}_2)$$

Independence rule is a special case with $\alpha = 0$

† *The Naive Bayes classifier has nothing to do with the Bayes classifier*

Shrinkage helps (true Σ is not diagonal)

Left two panels show the mean and variance of approximating $\eta^\top x$ by $\hat{\eta}^\top x$, and right two panels show the classification error. True Σ is an auto-regressive covariance matrix.



Guo et al. (2006)

Explanation: Bias-variance trade-off

$$\eta^\top X \sim \mathcal{N}\big(\pm\eta^\top\mu,\ \eta^\top\Sigma\eta\big), \qquad \delta(\eta) \equiv \frac{|\eta^\top\mu|}{\sqrt{\eta^\top\Sigma\eta}}$$

- For Fisher's and Independence rule ($\hat{\mu} \equiv \frac{\bar{X}_1 - \bar{X}_2}{2}$)

$$\hat{\delta}_{FR} \equiv \frac{\hat{\mu}^\top S^{-1}\mu}{\sqrt{\hat{\mu}^\top S^{-1}\Sigma S^{-1}\hat{\mu}}}, \quad \hat{\delta}_{IR} \equiv \frac{\hat{\mu}^\top D^{-1}\mu}{\sqrt{\hat{\mu}^\top D^{-1}\Sigma D^{-1}\hat{\mu}}}$$

- Population counterparts are ($D_0 \equiv \mathrm{diag}(\Sigma)$)

$$\delta_{FR} \equiv \sqrt{\mu^\top\Sigma^{-1}\mu}, \qquad \delta_{IR} \equiv \frac{\mu D_0^{-1}\mu}{\sqrt{\hat{\mu}^\top D_0^{-1}\Sigma D_0^{-1}\mu}}$$

Explanation: Bias-variance trade-off, II

- Loss of replacing FR by IR:

$$\delta_{IR} \quad < \quad \delta_{FR}$$

- Gain of replacing FR by IR:

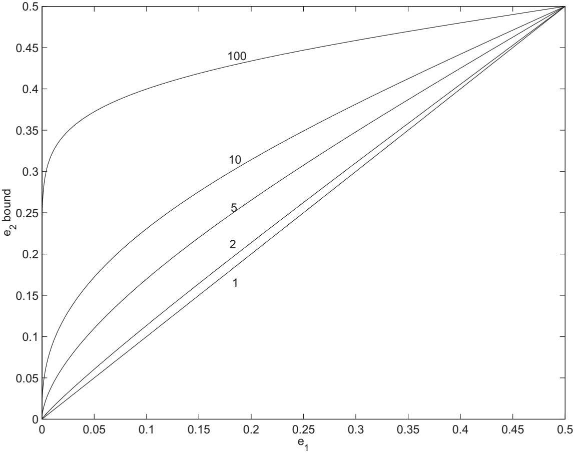
$$|\hat{\delta}_{IR} - \delta_{IR}| \quad \ll \quad |\hat{\delta}_{FR} - \delta_{FR}|$$

- Eventually, we bet on

$$\hat{\delta}_{IR} \quad > \quad \hat{\delta}_{FR}$$

Explanation: Bias-variance trade-off, III

x-axis: $\bar{\Phi}(\delta_{FR})$; y-axis: $\bar{\Phi}(\delta_{IR})$; $\bar{\Phi}$: tail CDF of $N(0,1)$; number on the curve is conditioning number of $D_0^{-\frac{1}{2}}\Sigma D_0^{-\frac{1}{2}}$; results suggest $\bar{\Phi}(\delta_{IR})$ is not much larger than $\bar{\Phi}(\delta_{FR})$ in many cases



Bickel and Levina (2004)

2020.11.20作业

- 使用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 和QDA 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。
- 使用美国国立癌症研究(NCI)规定的60种作为抗癌新药开发时必须筛查的癌细胞案例数据（数据是64×6830的表达式值矩阵，而labs是列出64种细胞系癌症类型的载体。）目标是用6830个基因对两种癌细胞进行分类：RENAL还是PROSTATE

  （1）从中通过描述统计选择出15个自变量，通过Fisher判别分别估计两类的均值和协方差矩阵，给出判别规则。

  （2）再通过IR-LDA方法，选择合适的α给出判别准则，比较(1)和(2)的结果。

Noise accumulation via too many features

- Big data: numerous features can be collected
- However, a large fraction of them have almost no contribution on differentiating two classes
- These useless features introduce huge noise

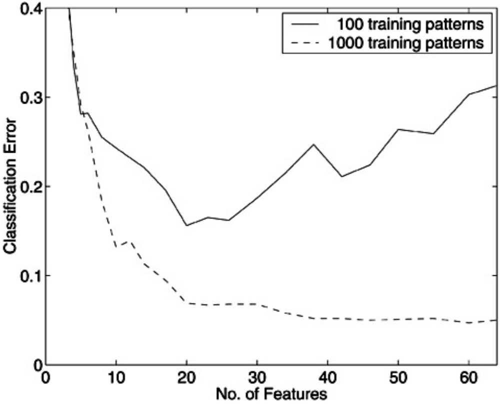


Figure is from Jain, Duin and Mao (2000)

Effect of noise accumulation in LDA

$$X_i \sim \mathcal{N}_p\big(y_i \cdot \mu,\ \Sigma\big), \quad y_i \in \{\pm 1\}, \quad i = 1, 2, ..., n$$
$$\text{useless feature} \Longleftrightarrow \text{zero entries of } \mu$$

The independence rule has an asymptotic classification error of 0.5 if

$$(\mu^\top D_0^{-1} \mu) \cdot \sqrt{n/p} \ \to \ 0, \quad \text{where } D_0 \equiv \text{diag}(\Sigma)$$

If μ has s nonzero entries each being $O(\tau)$, then the above situation happens when

$$\tau^2 \cdot \sqrt{ns^2/p} \ \to \ 0,$$

which can easily be true if $s \ll p$

Fan and Fan (2008)

Feature selection by two-sample t-test

$$T_j = \frac{\bar{X}_{1,j} - \bar{X}_{2,j}}{\sqrt{s_{1,j}^2/n_1 + s_{2,j}^2/n_2}}$$

- $s_{1,j}$, $s_{2,j}$: sample SD of feature j in two classes
- Rank features by $|T_j|$
- Select top m_0 features or features with $|T_j| > \alpha$ (m_0, α are chosen by cross-validation)
- Apply independence rule on selected features

Fan and Fan (2008)

Summary so far

- LDA framework for multivariate normal model
- It reduces to approximating $\eta = \Sigma^{-1}(\mu_1 - \mu_2)$ via training data
- HD Issue 1: Poor behavior of S^{-1}. Simple modification as shrinkage towards diagonal already significantly boosts performance
- HD Issue 2: Noise accumulation in too many features. Simple feature selection by 2-sample *t*-test already significantly boosts performance

Next, more careful thoughts of Issues 1&2 ...

Sparsity

- Sparsity of μ: Only a small fraction of features are useful
- Sparsity of Ω: Only a small fraction of pairs of variables are conditionally dependent

Remark. There's work assuming sparsity on $\eta \propto \Omega\mu$:

$$\hat{\eta} \in \arg\min_{\eta \in \mathbb{R}^p} \Big\{ \|\eta\|_1, \text{ subject to } \|S\eta - (\bar{X}_1 - \bar{X}_2)\|_\infty \le \lambda \Big\}$$

Intuition: $\eta = \Sigma^{-1}(\mu_1 - \mu_2) \quad \Leftrightarrow \quad \Sigma\eta = \mu_1 - \mu_2$

Cai and Liu (2011)

Sparse LDA for $\Sigma = I_p$

$$X_i \sim \mathcal{N}_p\big(y_i \cdot \mu,\ I_p\big), \quad y_i \in \{\pm 1\}, \quad i = 1, 2, ..., n$$

- Compute Z-scores: $Z = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (y_i \cdot X_i)$
- Thresholding: (feature selection & independence rule)

$$\hat{\eta}_{t,j} = \begin{cases} \mathrm{sgn}(Z_j), & |Z_j| \geq t, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \leq j \leq p$$

- Classify a new X by $\mathrm{sgn}(\hat{\eta}_t^\top X)$

Question: How to select the threshold? It is related to a deep question of *the goal of feature selection*

Ideal threshold

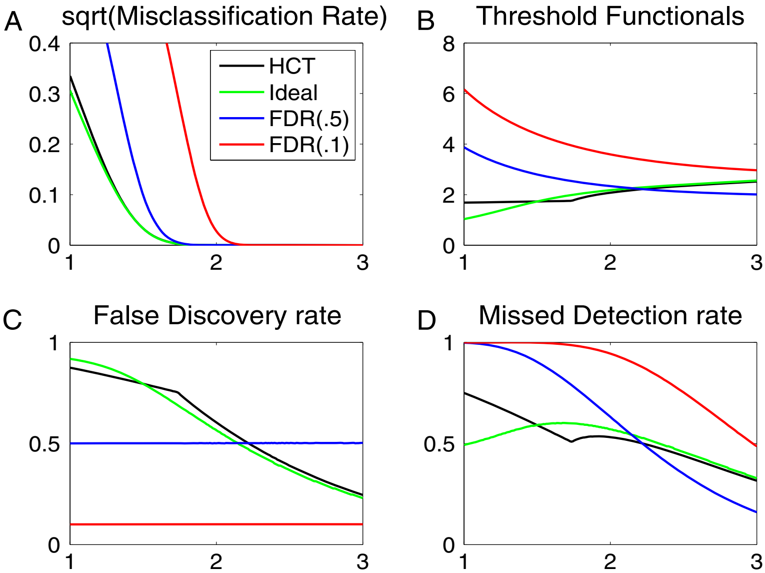
$$\hat{\delta}(t) = \hat{\delta}_t(\mu, \Sigma) \equiv \frac{\hat{\eta}_t^\top \mu}{\sqrt{\hat{\eta}_t^\top \Sigma \hat{\eta}_t}}$$

- Ideal threshold $\hat{t}^{ideal}$ is the maximizer of $\hat{\delta}(t)$
- Infeasible in practice, but provides insight of the ‘optimal’ threshold for classification purpose

Q: Good feature selection $\overset{?}{\Longleftrightarrow}$ good classification

Feature selection is analogous to signal recovery in Chapter 2. However, the optimal threshold is not data-driven. We instead use FDR threshold for feature selection purpose.

A simulation (x-axis: feature strength μ_j)



Donoho and Jin (2008)

[About 1% of features are useful]

Observations

- In the strong feature case, ideal threshold and FDR threshold are similar
- In the weak feature case, ideal threshold is much smaller than FDR threshold
- Ideal threshold maintains a low false negative rate; necessarily, it has to maintain a high false discovery rate in the *weak feature* case

A: Good feature selection $\neq$ good classification

When useful features are weak, it is impossible to separate them from useless ones. We set a low threshold to retain most useful features. Unavoidably, many useless features also retain.

A different thought

Classification is indeed a hypothesis test:

$$H_0 : X \sim \mathcal{N}_p(\mu, \Sigma) \quad v.s. \quad H_1 : X \sim \mathcal{N}_p(-\mu, \Sigma)$$

Data are X (testing) and $\{(X_i, y_i)\}_{i=1}^n$ (training)

- Z-scores summarize info. of training data
- Training data "changes the distribution of X"
- We instead view classification as a test of

$$X|Z \sim f_0(x; Z), \qquad v.s. \qquad X|Z \sim f_1(x; Z)$$

- This is a global testing (about all features)

A: Good global detection $=$ good classification

Higher Criticism Threshold (HCT)

$$Z = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (y_i \cdot X_i) \sim \mathcal{N}_p(\sqrt{n}\mu, I_p)$$

- Compute *P*-values: $\pi_j = \mathbb{P}(|Z_j| > \mathcal{N}(0, 1))$
- Sort: $\pi_{(1)} < \pi_{(2)} < \ldots < \pi_{(p)}$
- *HC*-scores: (slightly different from Chapter 2)

$$HC_{p,k} = \sqrt{p}\left[\frac{k/p - \pi_{(k)}}{\sqrt{(k/p)(1 - k/p)}}\right]$$

- HC-threshold (HCT):

$$t_p^{HC} = \hat{k}\text{-th largest absolute } Z\text{-score},$$

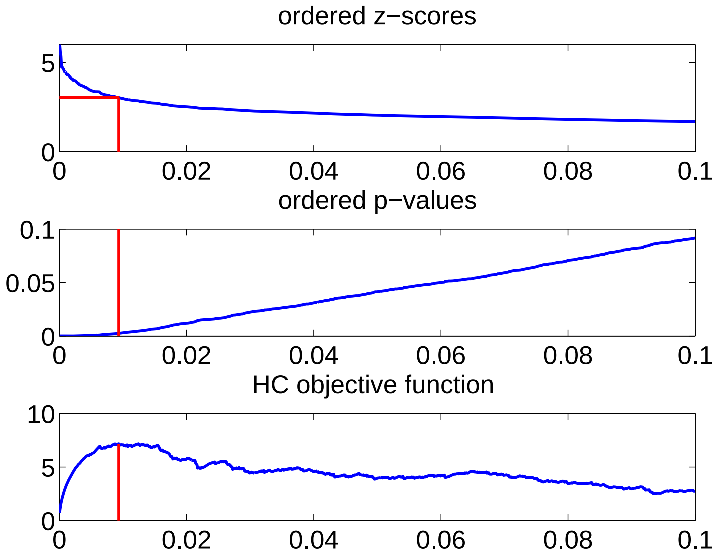
where $\hat{k} = \mathrm{argmax}_{\{1 \leq k \leq \frac{p}{10}\}}\{HC_{p,k}\}$

Comparison of two versions of HC

	Orthodox HC	New HC (HCT)
Objective	Global testing	Threshold choice for feature selection
$HC_{p,k}$	$\sqrt{p}\frac{k/p-\pi_{(k)}}{\sqrt{\pi_{(k)}(1-\pi_{(k)})}}$	$\sqrt{p}\frac{k/p-\pi_{(k)}}{\sqrt{(k/p)(1-k/p)}}$
Statistics	$\max_{\{1\leq k\leq \frac{p}{2}, \pi_{(k)}\geq \frac{\log(p)}{p}\}}\{HC_{p,k}\}$	$\hat{k}=\mathrm{argmax}_{\{1\leq k\leq \frac{p}{10}\}}\{HC_{p,k}\}$

Donoho and Jin (2008)

Illustration



Classification model

- Training samples $\{(X_i,\ y_i)\}_{i=1}^n$:

$$X_i|y_i \sim \mathcal{N}_p(y_i \cdot \mu,\ \Sigma), \quad y_i = \begin{cases} +1, & \text{w.p. } 1/2 \\ -1, & \text{w.p. } 1/2 \end{cases}$$

- Testing sample: $X|y \sim \mathcal{N}_p(y \cdot \mu,\ \Sigma)$
- Summarizing Z-scores:

$$Z = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (y_i \cdot X_i) \sim \mathcal{N}_p(\sqrt{n}\mu, \Sigma)$$

† Without loss of generality, we assume Σ has unit diagonals

Rare/Weak feature model

$$\sqrt{n} \cdot \mu_j \overset{iid}{\sim} (1-\epsilon)\nu_0 + \epsilon \cdot \nu_\tau, \qquad 1 \leq j \leq p$$

- As $p \to \infty$, parametrize (ϵ, τ) as

$$\epsilon_p = p^{-\beta}, \qquad \tau_p = \sqrt{2r \log p}, \qquad 0 < \beta, r < 1$$

- Link sample size n to p (3 types of growth):
  - (*No growth*): n is fixed
  - (*Slow growth*): $1 \ll n \ll p^\theta$, for any $\theta > 0$
  - (*Regular growth*): $n = p^\theta$ for some $\theta \in (0, 1)$

Donoho and Jin (2008), Jin (2009)

Two regions

The (β, r)-plane partitions into two regions:

- Possibility: misclassification error rate $\to 0$
- Impossibility: misclassification error rate $\to 1/2$

When $\Sigma = I_p$, the boundary between regions is

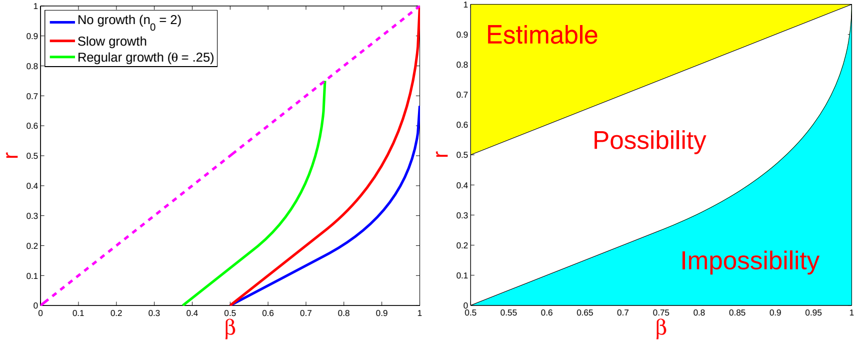
$$\rho_\theta(\beta) = \begin{cases} \frac{n}{n+1} \cdot \rho(\beta), & \text{no growth} \\ \rho(\beta), & \text{slow growth} \\ (1-\theta) \cdot \rho(\frac{\beta}{1-\theta}), & \text{regular growth} \end{cases}$$

where $\rho(\beta)$ is the detection boundary Chapter 2

$$\rho(\beta) = \begin{cases} 0, & 0 < \beta < 1/2 \\ (\beta - 1/2), & 1/2 \leq \beta < 3/4 \\ (1 - \sqrt{1-\beta})^2, & 3/4 \leq \beta < 1 \end{cases}$$

Jin (2009)

Phase Diagram ($\Sigma = I_p$)



Left: Classification Boundaries. Right: Detection boundary in Chapter 2

Insight of the classification boundary

Classification problem is indeed a testing problem

$$H_0 : X + \mu \sim \mathcal{N}_p(0, \Sigma) \quad v.s. \quad H_1 : X + \mu \sim \mathcal{N}_p(2\mu, \Sigma)$$

Additional data summarized in $Z \sim \mathcal{N}_p(\sqrt{n}\mu,\ \Sigma)$

- Z is unavailable: X contains $\approx p\epsilon_p$ useful features, each with a strength of $\mu_p = \tau_p/\sqrt{n}$
- Z is available: The posterior probability that X_j is a useful feature is not ϵ_p but $\epsilon_p(Z_j)$, where

$$\epsilon_p(z) = \frac{\epsilon_p \varphi(z - \tau_p)}{(1 - \epsilon_p)\varphi(z) + \epsilon_p \varphi(z - \tau_p)}$$

Effective β^* depends on (β, r, θ), so boundary alters

The case of a general Σ

The summarizing Z-scores

$$Z = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (y_i \cdot X_i) \sim \mathcal{N}_p(\sqrt{n}\mu,\ \Sigma)$$

- Stein's normal means model with colored noise
- The Innovated transformation boosts marginal SNR simultaneously at all sites

$$Z \qquad \mapsto \qquad \Omega Z$$

- If Ω is sparse, can estimate it via training data

Sparse LDA for a sparse Ω

$$X_i \sim \mathcal{N}_p\big(y_i \cdot \mu,\ \Omega^{-1}\big), \quad y_i \in \{\pm 1\}, \quad i = 1, 2, ..., n$$

- Obtain an estimate of Ω from $\{(X_i, y_i)\}_{i=1}^n$
- Compute Z-scores: $Z = \frac{1}{\sqrt{n}} \sum_{i=1}^n (y_i \cdot X_i)$
- Boost SNR by Innovated Transform: $\widetilde{Z} = \hat{\Omega} Z$
- Compute P-values: $\pi_j = \mathbb{P}\big(|\widetilde{Z}_j| > [\hat{\Omega}_{jj}]^{\frac{1}{2}}\big)$
- Use P-values to compute HCT $\hat{t}_{HC}$
- Thresholding:

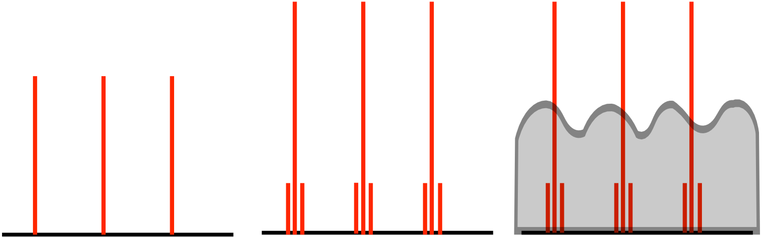
$$\hat{\eta}_{HCT,j} = \begin{cases} \mathrm{sgn}(\widetilde{Z}_j), & |\widetilde{Z}_j| \geq \hat{t}_{HC}, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \leq j \leq p$$

- Classify a new X by $\mathrm{sgn}(\hat{\eta}_{HCT}^{\top} X)$

In RW settings, $\hat{\eta}_t$ tries to estimate μ

- (A). Treat $\hat{\eta}_t$ as an estimator of μ
- (B). Treat $\hat{\eta}_t$ as an estimator of $\Omega\mu$

Idea. $\mu \mapsto \Omega\mu$ produces (1) main signals and (2) many weaker signals, where (2) are impossible to estimate; you may think you are estimating $\Omega\mu$, but you are actually estimating μ



4.3 判别式模型的基本代表-Logistic回归

假设连续情形下，

- 对于二分类问题$\{C_1, C_2\}$而言，

$$p(C_1|x) = \frac{p(x, C_1)}{p(x)} = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)}$$
$$= \frac{1}{1 + \exp(-a(x)})) = \sigma(a);$$

- 对数优势比函数(log odds)：$a(x) = \ln \frac{p(C_1|x)}{p(C_2|x)}$;
- $\sigma(a)$逻辑挤压变形(Logistic sigmoid)函数：$\sigma(a) = \frac{1}{1+\exp(-a)}$;
- $\sigma(a)$的性质：对称性$\sigma(-a) = 1 - \sigma(a)$; $a = \ln(\frac{\sigma}{1-\sigma})$;
- 对于多分类问题$K > 2, \{C_1, ..., C_K\}, k\epsilon 1, ..., K$ 而言，

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{\sum p(x|C_j)p(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)};$$

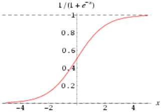
- $a_k = \ln p(x|C_k)p(C_k)$;

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$\beta=(w,b),\tilde{x}_i=(x_i,1)$,那么

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

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

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

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

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

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

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

- 注意:以上求导过程中，$\ln(1 + \beta^T\tilde{x}_i)$对$\beta_j$求导

  时，$p(x_i; \beta) = \frac{e^{\beta^T\tilde{x}}}{1+e^{\beta^T\tilde{x}}}$
- 特别的，

$$\frac{\partial^2 l(\beta)}{\partial \beta_0 \partial \beta_j^T} = \sum_{i=1}^{m} \tilde{x}_i p_1(\tilde{x}_i; \beta)(1 - p_1(\tilde{x}_i; \beta))$$

-

$$\frac{\partial^2 l(\beta)}{\partial \beta_0 \partial \beta_0^T} = \sum_{i=1}^{m} p_1(\tilde{x}_i; \beta)(1 - p_1(\tilde{x}_i; \beta))$$

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

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

- 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$p_k = -\nabla f(x^{(k)}), \lambda_k$ 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

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

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

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

- 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：$\beta^{t+1} = \beta^t - \frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}^{-1} \frac{\partial l(\beta)}{\partial\beta}$
- 其中关于β的一阶二阶导数分别为：

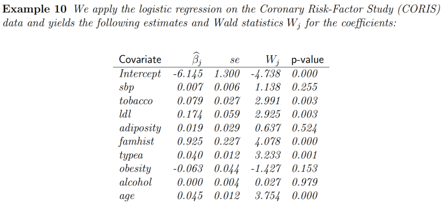
$$\frac{\partial l(\beta)}{\partial\beta_j} = \sum_{i=1}^{m} \tilde{x}_{ij}(y_i - p_1(\tilde{x}_i;\beta));$$

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

- 特别的，

$$\frac{\partial^2 l(\beta)}{\partial\beta_0\partial\beta_j^T} = \sum_{i=1}^{m} \tilde{x}_i p_1(\tilde{x}_i;\beta)(1 - p_1(\tilde{x}_i;\beta))$$

- $$\frac{\partial^2 l(\beta)}{\partial\beta_0\partial\beta_0^T} = \sum_{i=1}^{m} p_1(\tilde{x}_i;\beta)(1 - p_1(\tilde{x}_i;\beta))$$



Example 10 *We apply the logistic regression on the Coronary Risk-Factor Study (CORIS) data and yields the following estimates and Wald statistics W_j for the coefficients:*

Covariate	$\widehat{\beta}_j$	*se*	W_j	p-value
Intercept	*-6.145*	*1.300*	*-4.738*	*0.000*
sbp	*0.007*	*0.006*	*1.138*	*0.255*
tobacco	*0.079*	*0.027*	*2.991*	*0.003*
ldl	*0.174*	*0.059*	*2.925*	*0.003*
adiposity	*0.019*	*0.029*	*0.637*	*0.524*
famhist	*0.925*	*0.227*	*4.078*	*0.000*
typea	*0.040*	*0.012*	*3.233*	*0.001*
obesity	*-0.063*	*0.044*	*-1.427*	*0.153*
alcohol	*0.000*	*0.004*	*0.027*	*0.979*
age	*0.045*	*0.012*	*3.754*	*0.000*

Logistic回归的做法步骤总结：

- （表示）取一个二项分布的似然函数，再最大似然函数，转换成求最小二乘法;
- （求解）再求导出β向量的解析解，用梯度下降，牛顿等方法估计参数β的最优解;
- （不足）线性模型重点就是放在求某个参数上。但是，当数据量少的时候，logistic回归的点估计很容易造成过拟合overfitting。
- （解决）贝叶斯估计：估计目标是一个分布。而不是一个最优化的值β_{MAP})，通过似然函数× 先验求出后验概率分布之后，再用它去积分进行类别预测，考虑的是全局的所有β，这样就可以消除过拟合。

迭代重加权平方Iterative reweighted least squares(IRIS)

设计用牛顿法优化。先计算Hessian矩阵

$$\mathbf{H} = \nabla^2 E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^T = \Phi^T\mathbf{R}\Phi$$

其中$\mathbf{R}$是对角矩阵，$\mathbf{R}_{nn} = y_n(1 - y_n)$. 注意$\mathbf{H}$正定.

$$\mathbf{w}^{(new)} = \mathbf{w}^{(old)} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) = (\Phi^T\mathbf{R}\Phi)^{-1})\Phi^T\mathbf{R}\mathbf{z}$$

其中

$$\mathbf{z} = \Phi\mathbf{w}^{(old)} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$$

注意，每次迭代，$\mathbf{R}$要重新计算，实际上R可以被解释为预测值的协方差

$$\mathbb{E}[t] = \sigma(\mathbf{w}^T\phi) = y$$
$$var[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{w}^T\phi) - \sigma(\mathbf{w}^T\phi)^2 = y(1 - y)$$

标准链接函数Canonical link functions

这里的指数族分布是关于t的

- $$p(t|\eta, s) = \frac{1}{s}h(\frac{t}{s})g(\eta)\exp\{\frac{\eta t}{s}\}$$
- 根据第2章的结论
  $$y = \mathrm{E}[t|\eta] = -s\frac{d}{d\eta}\ln g(\eta)$$
- y和η有关系，记为$\eta(y)$,其中$y = f(\mathbf{w}^T\phi)$
  $$\ln p(\mathbf{t}|\eta, s) = \sum_{n=1}^{N}\{\ln g(\eta_n) + \frac{\eta_n t_n}{s}\} + \mathrm{const}$$
- $$\nabla_{\mathbf{w}}\ln p(\mathbf{t}|\eta, s) = \sum_{n=1}^{N}\{\frac{d}{d\eta_n}\ln g(\eta_n) + \frac{t_n}{s}\}\frac{d\eta_n}{dy_n}\frac{dy_n}{d(\mathbf{w}^T\phi_n)}\nabla(\mathbf{w}^T\phi_n))$$
- $$\nabla_{\mathbf{w}}\ln p(\mathbf{t}|\eta, s) = \sum_{n=1}^{N}\{\frac{1}{s}t_n - y_n\}\eta'(y_n)f'(\mathbf{w}^T\phi_n)^T\phi_n$$

数据标注出错的情况-weak-label

如果$f^{-1}=\eta$,那么$\eta'(y_n)f'(\mathbf{w}^T\phi_n)=1$,注意此时，$\eta=\mathbf{w}^T\phi$得到

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

高斯函数$s=\beta^{-1}$.对于逻辑回归$s=1$.
如果二分类数据集以ϵ的概率标错，则预测概率可以建模为

$$p(t|x)=(1-\epsilon)\sigma(x)+\epsilon(1-\sigma(x))=\epsilon+(1-2\epsilon)\sigma(x)$$

这里ϵ可以是参数或超参数。这很像Label-Noise Robust Generative Adversarial Networks论文的基础版

Laplace近似

- logistc回归的贝叶斯理论中，后验分布不再是高斯分布，这样就不能精确的对w求积分来解决估计问题，而是有必要做某种形式的近似。基于这样一种原因引入拉普拉斯近似。拉普拉斯近似就是用来近似分布。
- 找到定义在一组变量上的概率密度的高斯近似。
- 拉普拉斯近似的推导：
  - 1.寻找后验分布的众数，假设分布:

    $$p(z) = \frac{f(z)}{Z},$$

    Z是是归一化系数。假定Z的值是未知的。在拉普拉斯方法中，要寻找高斯近似$q(z)$，它的中心位于$p(z)$的众数位置，寻找众数就是要寻找一个点使$p'(z) = 0$.
  - 2.对其对数泰勒展开；高斯分布的对数是变量的二次函数。所以考虑$lnf(z)$以众数z_0 为中心的泰勒展开:

    $$\ln(z) \simeq \ln f(z_0) - \frac{1}{2}A(z - z_0)^2;$$

    没有一阶项是因为z_0是概率分布的局部最大值. 两边取指数:

Fig4-14a.png

- $$f(x) \simeq f(z_0)\exp\{-\frac{A}{2}(z-z_0)^2\}$$
- 3.归一化:使用归一化的高斯分布的标准形式，得到归一化的概率分布$q(z)$：

$$q(z) = (\frac{A}{2\pi})^{\frac{1}{2}}\exp\{-\frac{A}{2}(z-z_0)^2\}$$

- 高斯近似只在精度$A > 0$时有良好的定义，也就是驻点z_0一定是个局部最大值，使得$f(z)$在驻点z_0 处的二阶导数为负.

推广到M维空间z上

- M 维空间z上的概率分布$p(z)=f(z)Z$,在驻点z_0处，梯度$\nabla f(z)$将会消失，在驻点z_0处展开，我们有：

$$\ln(z)=\ln f(z_0)-\frac{1}{2}(z-z_0)^T A(z-z_0);$$

其中$M\times M$的Hessian矩阵定义为$A=-\nabla\nabla\ln f(z)|_{z=z_0}$;

- 同时取指数：

$$f(z)\simeq f(z_0)\exp\{-\frac{1}{2}(z-z_0)^T A(z-z_0)^2\}$$

- 归一化，$q(z)=\mathcal{N}(z,z_0,A)$,$q(z)$正比于$f(z)$:

$$\frac{|A|^{\frac{1}{2}}}{2\pi^{\frac{M}{2}}}\exp\{-\frac{1}{2}(z-z_0)^T A(z-z_0)^2\}$$

- 前提是精度矩阵A是正定的,表明驻点z_0一定是一个局部最大值.

数据越多，拉普拉斯近似会更有用（根据中心极限定理，分布会更趋近于高斯）

4.5.1拉普拉斯近似-模型比较和BIC

对于模型证据

$$p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)d\theta$$

如果取$f(\theta) = p(\mathcal{D}|\theta)p(\theta)$，$Z = p(\mathcal{D})$，要拟合的分布为$p(\theta|\mathcal{D})$，则

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\theta_{MAP}) + \ln p(\theta_{MAP}) + \frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{A}| \tag{1}$$

右侧其中后三项叫做**Occam因子Occam factor**，惩罚模型复杂度
如果认为参数先验非常宽，并且$\mathbf{A}$满秩，那么可以非常粗略认为

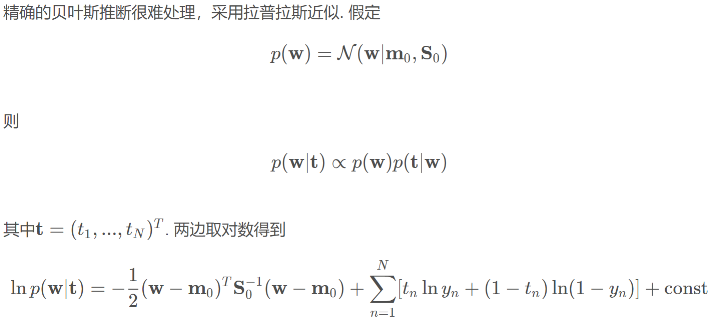
$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\theta_{MAP}) - \frac{1}{2}M\ln N$$

其中N是样本数，M是参数量，这就是很多人都知道的**贝叶斯信息准则Bayesian Information Citerion(BIC)**

- 相比于AIC，BIC更严重地惩罚了模型复杂度
- 不过实际使用中，AIC和BIC虽然容易估计。但Hessian矩阵往往不满秩，导致结果不正确

Laplace在Bayesian Logistic Regression上的应用1/4

对于logistic回归，精确地贝叶斯推断是无法处理的，特别的，计算后验概率分布需要对先验概率分布于似然函数的乘积进行归一化，而似然函数本身由一系列的logistic sigmoid函数的乘积组成，每个数据点都有一个logistic sigmoid函数。



精确的贝叶斯推断很难处理，采用拉普拉斯近似. 假定

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

则

$$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{w})p(\mathbf{t}|\mathbf{w})$$

其中$\mathbf{t} = (t_1, ..., t_N)^T$. 两边取对数得到

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{1}{2}(\mathbf{w}-\mathbf{m}_0)^T\mathbf{S}_0^{-1}(\mathbf{w}-\mathbf{m}_0) + \sum_{n=1}^{N}[t_n \ln y_n + (1-t_n)\ln(1-y_n)] + \text{const}$$

Laplace在Bayesian Logistic Regression上的应用2/4

二阶导数为

$$\mathbf{S}_N = -\nabla^2 \ln p(\mathbf{w}|\mathbf{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T$$

所以对应后验为

$$q(\mathbf{w}) = (\mathbf{w}|\mathbf{w}_{MAP}, \mathbf{S}_N)$$

Bayesian Logistic Regression:预测分布3/4

进行预测

$$
\begin{aligned}
p(\mathcal{C}_1|\phi,\mathbf{t}) &= \int p(\mathcal{C}_1|\phi,\mathbf{w})p(\mathbf{w}|\mathbf{t})d\mathbf{w} \simeq \int \sigma(\mathbf{w}^T\phi)q(\mathbf{w})d\mathbf{w} \\
&= \int \left[\int \delta(a-\mathbf{w}^T\phi)\sigma(a)da\right] q(\mathbf{w})d\mathbf{w} = \int \sigma(a)p(a)da
\end{aligned}
$$

时，

其中

$$p(a) = \int \delta(a - \mathbf{w}^T\phi)q(\mathbf{w})d\mathbf{w}$$

注意a是$\mathbf{w}$的线性映射，q是高斯分布，所以a也是高斯分布

而且

$$\mu_a = \mathbb{E}[a] = \int p(a)ada = \int q(\mathbf{w})\mathbf{w}^T\phi d\mathbf{w} = \mathbf{w}_{MAP}^T\phi$$

$$\sigma_a^2 = var[a] = \int p(a)\{a^2 - \mathbb{E}[a]^2\}da = \int q(\mathbf{w})\{(\mathbf{w}^T\phi)^2 - (\mathbf{w}_{MAP}^T\phi)^2\}d\mathbf{w} = \phi^T\mathbf{S}_N\phi$$

Bayesian Logistic Regression:预测分布4/4

从而

$$p(\mathcal{C}_1|\phi,\mathbf{t}) = \int \sigma(a)\mathcal{N}(\mu_a,\sigma_a^2)da$$

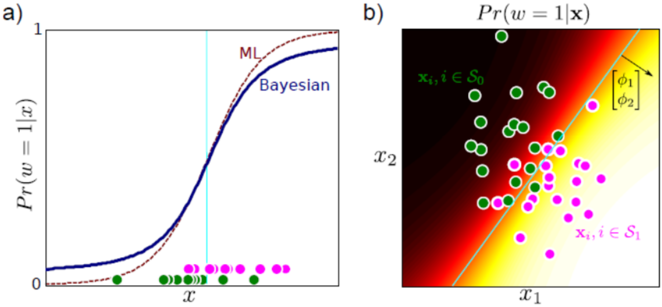
该式无法解析解，不过可以用probit函数$\Phi(\lambda a)$近似代替$\sigma(a)$，其中$\lambda^2=\pi/8$，这样可以求出解析解为

$$\int \Phi(\lambda a)\mathcal{N}(a|\mu_a,\sigma_a^2)da = \Phi\left(\frac{\mu_a}{(\lambda^{-2}+\sigma_a^2)^{1/2}}\right)$$

再次利用近似式子，带回sigmoid函数，得到$\Phi\left(\frac{\mu_a}{(\lambda^{-2}+\sigma_a^2)^{1/2}}\right) \approx \sigma(\kappa(\sigma^2)\mu)$，其中$\kappa(\sigma_a^2)=(1+\pi\sigma_a^2/8)^{-1/2}$

$$p(\mathcal{C}_1|\phi,t) \approx \sigma(\kappa(\sigma_a^2)\mu_a)$$

- 这里的决策边界是$\mu_a = w_{MA}^T\phi = 0$，与MAP一致。但在远处的概率比MAP更加温和



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

3.5 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:

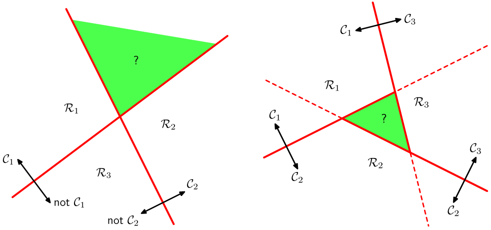
- 在降维过程中可以使用类别的先验知识经验;
- LDA在样本分类信息依赖均值而不是方差。

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。

Discriminant functions for $N > 2$ classes

- One possibility is to use N two-way discriminant functions. Each function discriminates one class from the rest.
- Another possibility is to use $N(N-1)/2$ two-way discriminant functions,Each function discriminates between two particular classes.
- Both these methods have problems



A simple solution

- 使用K个分类函数$g_1(x), ..., g_k(x)$,每个线性判别函数有下面的形式

$$g_k(x) = w_k^T x + w_{k0}$$

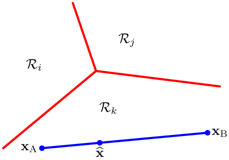


Figure: 如果两个点在同一个决策区域，决策区域满足单连通和凸性质

- 如果$g_k(x_A) > g_l(x_A), g_k(x_B) > g_l(x_B), \forall \hat{x} = \lambda x_A + (1 - \lambda) x_B, 0 \leq \lambda \leq 1,$

$$g_k(\hat{x}) > g_l(\hat{x})$$

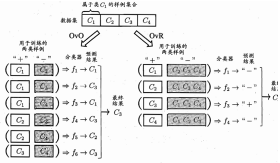
The simplest representation of a linear discriminant function can be expressed as:

$$y(x) = f(\omega^T x + \omega_0)$$

在机器学习文献中，f(.)被称为激活函数。它的反函数是链接函数(link function). 决策面对应于$y(x) = const$. The normal distance from the origin to the decision surface is given by

$$\frac{\omega^T x}{\|\omega\|} = -\frac{\omega_0}{\|x\|}$$

- 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。

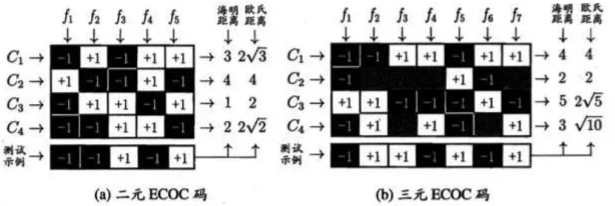


属于类C_1的样例集合
数据集
C_1 C_2 C_3 C_4
OvO
OvR
用于训练的两类样例
“+” “−”
分类器
预测结果
$(C_1\ C_2) \Rightarrow f_1 \rightarrow C_1$
$(C_1\ C_3) \Rightarrow f_2 \rightarrow C_3$
$(C_1\ C_4) \Rightarrow f_3 \rightarrow C_1$
$(C_2\ C_3) \Rightarrow f_4 \rightarrow C_3$
$(C_2\ C_4) \Rightarrow f_5 \rightarrow C_2$
$(C_3\ C_4)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

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$y = w^Tx + b$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？