机器学习

Chapter 1

线性模型

给定一个输入向量 $(x_1, x_2, ..., x_d)$，我们想获得一个输出 y。线性回归模型试图学得一个通过属性的线性组合来进行预测的函数，即

$$\begin{aligned} f(\mathbf{x}) &= w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b \\ &= \mathbf{w}^T \mathbf{x} \end{aligned} \tag{1.1}$$

这里 $\mathbf{w} = (b, w_1, w_2, \cdots, w_d)$ 是待学习的未知参数，变量 $\mathbf{x} = (1, x_1, x_2, ..., x_d)$ 可以有不同来源：

- quantitative inputs;
- transformations of quantitative inputs, such as log, square-root or square;
- basis expansions, such as $x_2 = x_1^2$, $x_3 = x_1^3$, leading to a polynomial representation;
- numeric or "dummy" coding of the levels of qualitative inputs. For example, if G is a five-level factor input, we might create x_j, $j = 1, \ldots, 5$, such that $x_j = I(G = j)$.
- interactions between variables, for example, $x_3 = x_1 \cdot x_2$.

更一般地，考虑单调可微函数 $g(\cdot)$，令 $g(y) = \mathbf{w}^T \mathbf{x} + b$，这样 $\mathbf{w}^T \mathbf{x} + b$ 可以逼近一个 $g(\cdot)$，于是可以使用线性模型去逼近不同的函数，称这样的模型为**广义线性模型**（generalized linear model），即：

$$y = g^{-1}(\mathbf{w}^T \mathbf{x} + b) \tag{1.2}$$

其中，称 $g(\cdot)$ 为**联系函数**（link function）。显然，对数线性回归是广义线性回归模型在 $g(\cdot) = ln(\cdot)$ 时的特例。

一般线性模型是求输入空间到输出空间的线性函数映射，广义线性模型实际上是求输入空间到输出空间的非线性函数映射，而联系函数起到了将线性回归模型的预测值和真实标记联系起来的作用。

1.1. 线性回归模型

1.1.1. 线性回归

线性回归（linear regression）是一种回归分析技术。给定数据集，线性回归试图学习到一个线性模型以尽可能准确地预测实值输出标记。通过在数据集上建立线性模型，建立代价函数（loss function），最终以优化代价函数为目标确定模型参数，从而得到模型用以后续的预测。

基于均方误差或残差平方和最小化来进行模型求解的方法称为**最小二乘法**（least square method）。在线性回归中，最小二乘法就是试图找到一条直线，使得所有样本到直线上的欧式距离之和最小。

先考虑单元线性回归，对单个输入向量 $(x_1, x_2, ..., x_d)$ ，则参数 **w** 的**均方误差** (square loss) 为：

$$\begin{aligned} SL(w) &= \sum_{i=1}^{N}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N}(y_i - wx_i - b)^2 \end{aligned} \tag{1.3}$$

现求解 w 使得 $SL(w)$ 最小化。将 $SL(w)$ 对 w 和 b 分别求导得

$$\frac{\partial SL(w)}{\partial w} = 2\left(w\sum_{i=1}^{m}x_i^2 - \sum_{i=1}^{m}(y_i - b)x_i\right)$$

$$\frac{\partial SL(w)}{\partial w} = 2\left(mb - \sum_{i=1}^{m}(y_i - wx_i)\right)$$

令上面两个式子为零可得到 w 和 b 最优解的闭式解：

$$w = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2} \tag{1.4}$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - mx_i) \tag{1.5}$$

其中 $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$ 为 x 的均值。

现考虑多元线性回归。在整个数据集 D 上，记矩阵 $\mathbf{X}_{N\times(d+1)}$ 为输入的向量，记 $\mathbf{y}_{N\times 1}$ 为训练集的输出，参数记为 $\mathbf{w}_{(d+1)\times 1}$。则 $\mathbf{w}_{(d+1)\times 1}$ 的**残差平方和**（residual sum of squares）为：

$$\begin{aligned} RSS(\mathbf{w}) &= \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})(\mathbf{y} - \mathbf{X}\mathbf{w})^T \end{aligned} \tag{1.6}$$

对 **w** 求导，得到：

$$\frac{\partial RSS}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\frac{\partial RSS}{\partial \mathbf{w}\partial \mathbf{w}^T} = -2\mathbf{X}^T\mathbf{X}$$

1. 当 **X** 为满秩矩阵时，则 $\mathbf{X}^T\mathbf{X}$ 是正定矩阵，令上式为零

   $$\mathbf{X}^T\mathbf{X} = 0 \tag{1.7}$$

   得到 **w** 的解为：

   $$\hat{\mathbf{w}} = \mathbf{X}^T\mathbf{X}^{-1}\mathbf{X}^T\mathbf{y} \tag{1.8}$$

   因而，

   $$\hat{\mathbf{y}} = X\hat{\mathbf{w}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1.9}$$

   $\hat{\mathbf{y}}$ 是 **y** 在 **X** 的列空间上的正交映射。这个正交性用公式 (1.7) 表示。矩阵 $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ 计算正交映射，因此它也被称为映射矩阵（projection matrix）。
2. 如果 **X** 不是满秩矩阵时，存在多个解析解，它们都能使得损失函数最小化，选择哪一个解析解作为输出，由学习算法的归纳偏好决定。这时候常常引入正则化（regularization）项。常见的正则化项如 L_1 正则化或 L_2 正则化。

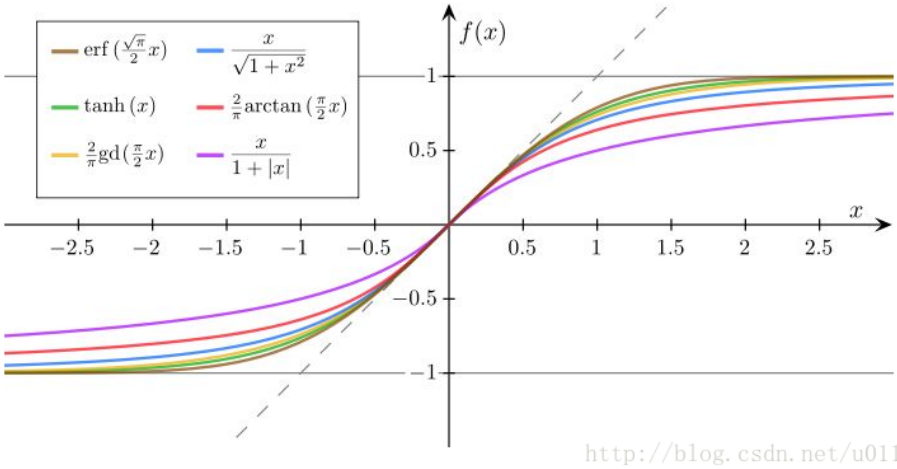

Figure 1. 阶跃函数近似的函数

(a) L_1 正则化：

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{argmin}\left[\mathbf{y} - \mathbf{X}\hat{\mathbf{w}} + \lambda||\mathbf{w}||\right](\lambda > 0) \tag{1.10}$$

(b) L_2 正则化：

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{argmin}\left[\mathbf{y} - \mathbf{X}\hat{\mathbf{w}} + \lambda||\mathbf{w}||^2\right](\lambda > 0) \tag{1.11}$$

1.1.1.1. 正则化问题

1.1.2. logistic 回归（logistic regression）

前面介绍了使用线性模型进行回归学习，下面则使用线性模型做分类。考虑到在广义线性模型中，只需找到一个单调可微函数将分类任务的真实标记和线性回归模型的预测值联系起来，就可以用线性模型完成分类任务了。

考虑二分类任务，其输出标记 $y \in \{0,1\}$，而线性回归模型产生的预测值 $z = \mathbf{w}^T\mathbf{x} + b$ 是实值，于是，我们需要将实值 □□□{0,1}$ 值，最理想的函数则是 Heaviside 阶跃函数（Heaviside step function）：

$$\begin{aligned} H(x) &= \frac{d}{dx}max\{x,0\} \\ &= \begin{cases} 0, x < 0 \\ 1, x > 0 \end{cases} \end{aligned} \tag{1.12}$$

但 Heaviside 阶跃函数不连续，故不可导。我们需要找到一个类似的函数，并且连续可导，常用的便是 **logistic 函数**（logistic function），也称为 Sigmoid 函数：

$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} = \frac{1}{2} + \frac{1}{2}tanh(\frac{x}{2}) \tag{1.13}$$

为什么要选用 logistic 函数，难道就是因为它连续可微和阶跃函数很像就足够了吗？下面这些函数也有类似的性质：
这个其实是考虑到 logistic 函数的导函数，我们看一下它的导函数形式：

$$\frac{d}{dx}f(x) = f(x)(1 - f(x)) \tag{1.14}$$

logistic 函数的导函数可以直接通过原函数很方便的计算出来，做参数估计时是需要作求导操作，函数有这样的特性是非常方便的。也就是它在众多函数中脱颖而出的原因之一。此外，logistic 函数的逆函数是对数几率函数（logit function），即：

$$logistic^{-1}(x) = logit(x) = log(\frac{x}{1-x}) \tag{1.15}$$

对数几率函数即几率 (odds) 的对数。几率（odds）指一个事件发生的概率与该事件不发生的概率的比值，设事件发生的概率为 x，则：

$$odds(x) = \frac{x}{1-x} \tag{1.16}$$

因此，logistic 回归实际上是用线性回归模型的预测结果去逼近真实标记的对数几率，此时联系函数为 $h(x) = logit(x) = log(\frac{x}{1-x})$，故 logistic 回归也叫**对数几率回归**。此时，训练的广义线性模型为：

$$E(y_i|\mathbf{x}_i) = g^{-1}(\mathbf{w}^T\mathbf{x} + b) = logistic(\mathbf{w}^T\mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}} \tag{1.17}$$

如果表示成条件概率分布函数，则公式为：

$$P(y_i = y|\mathbf{x}_i) = p_i^y(1-p_i)^{1-y} = \frac{e^{(\mathbf{w}x_i+b)\cdot y}}{1 + e^{\mathbf{w}\cdot x_i}} \tag{1.18}$$

1.1.2.1. logistic 回归的参数估计

现在，需要确定式（1.17）中的 **w** 和 b。为便于讨论，记 $\beta = \{\mathbf{w}; b\}$。我们可以通过**极大似然法**来估计回归系数 β。

对于给定的训练数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$，其中 $\mathbf{x}_i \in \mathbb{R}^n$，$y_i \in \{0, 1\}$，二项 logistic 回归模型的似然函数为：

$$\prod_{i=1}^{N} [P(y_i = 1|\mathbf{x}_i)]^{y_i} [1 - P(y_i = 1|\mathbf{x}_i)]^{1-y_i} \tag{1.19}$$

对数似然函数为：

$$\begin{aligned}
\ell(\beta) &= \sum_{i=1}^{N} logp(y_i|\mathbf{x}_i; \beta) \\
&= \sum_{i=1}^{N} [y_i logp(y_i = 1|\mathbf{x}_i) + (1 - y_i) logp(y_i = 0|\mathbf{x}_i)] \\
&= \sum_{i=1}^{N} \left[y_i log \frac{p(y_i = 1|\mathbf{x}_i)}{p(y_i = 0|\mathbf{x}_i)} + logp(y_i = 0|\mathbf{x}_i) \right] \\
&= \sum_{i=1}^{N} \left[y_i \cdot \beta \mathbf{x}_i^T - log\left(1 + e^{\beta \mathbf{x}_i^T}\right) \right]
\end{aligned} \tag{1.20}$$

然后对 $\ell(\beta)$ 求极大值。此式是关于 β 的高阶可导连续凸函数，根据凸优化理论，经典的数值优化算法，如梯度下降法、牛顿法等都可以求得其最优解，于是，可以得到：

$$\hat{\beta} = \underset{\beta}{argmin}(l(\beta)) \tag{1.21}$$

那么学到的二项 logistic 回归模型为：

$$E(y = 0|\mathbf{x}) = \frac{1}{1 + e^{\hat{\beta}\cdot\mathbf{x}}} \tag{1.22}$$

$$E(y = 1|\mathbf{x}) = \frac{e^{\hat{\beta}\cdot\mathbf{x}}}{1 + e^{\hat{\beta}\cdot\mathbf{x}}} \tag{1.23}$$

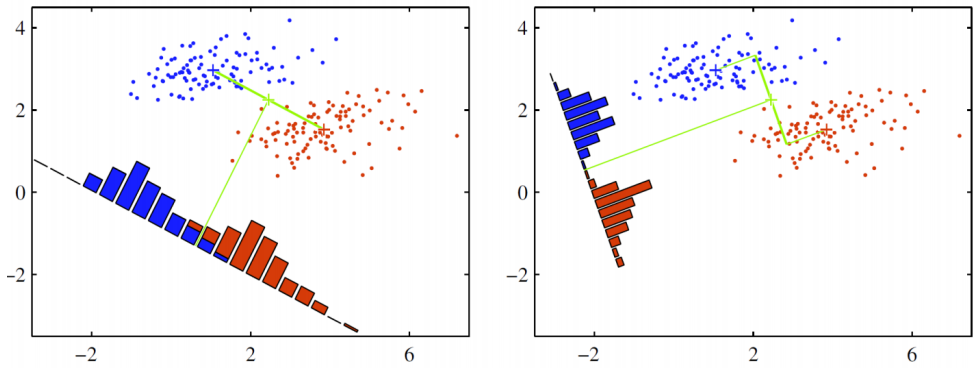

Figure 2. LDA 在不同方向的投影

梯度下降法

牛顿法

1.2. 线性判别分析

在学习**线性判别分析**（Linear Discriminant Analysis，LDA）之前，有必要将其自然语言处理领域的 LDA 区别开来。在自然语言处理领域，LDA 是**隐含狄利克雷分布**（Latent Dirichlet Allocation，LDA），它是一种处理文档的主题模型。我们本文只讨论线性判别分析，因此后面所有的 LDA 均指线性判别分析。

LDA 是一种监督学习的降维技术，也就是说它的数据集的每个样本是有类别输出的。这点和 PCA 不同。PCA 是不考虑样本类别输出的无监督降维技术。LDA 的思想为：

- 训练时：设法将训练样本投影到一条直线上，使得同类样本的投影点尽可能地接近而异类的样本投影点尽可能的远；
- 预测时：将待预测样本投影到学到的直线上，根据它的投影点的位置判断类别；

假设有两类数据分别为红色和蓝色，如下图所示，这些数据特征是二维的，我们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的大。
从直观上可以看出，右图要比左图的投影效果好，因为右图的黑色数据和蓝色数据各个较为集中，且类别之间的距离明显。左图则在边界处数据混杂。以上就是 LDA 的主要思想了，当然在实际应用中，数据是多个类别的，数据一般也是超过二维的，投影后的也一般不是直线，而是一个低维的超平面。

1.2.1. 瑞利商（Rayleigh quotient）与广义瑞利商（genralized Rayleigh quotient）

瑞利商是指这样的函数 $R(\mathbf{A}, \mathbf{x})$:

$$R(\mathbf{A}, \mathbf{x}) = \frac{\mathbf{x}^H \mathbf{A} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \tag{1.24}$$

其中 $\mathbf{x}$ 为非零向量，而 $\mathbf{A}$ 为 $n \times n$ 的 Hermitan 矩阵。Hermitan 矩阵就是满足共轭转置矩阵和自己相等的矩阵，即 $\mathbf{A}^H = \mathbf{A}$。如果矩阵 $\mathbf{A}$ 是实矩阵，则满足 $\mathbf{A}^T = \mathbf{A}$ 的矩阵即为 Hermitan 矩阵。

瑞利商 $R(\mathbf{A}, \mathbf{x})$ 有一个非常重要的性质，即它的最大值等于矩阵 $\mathbf{A}$ 最大的特征值，而最小值等于矩阵 $\mathbf{A}$ 的最小的特征值，也就是满足：

$$\lambda_{min} \leq \frac{\mathbf{x}^H \mathbf{A} \mathbf{x}}{\mathbf{x}^H \mathbf{x}} \leq \lambda_{max} \tag{1.25}$$

当向量 $\mathbf{x}$ 是标准正交基时，即满足 $\mathbf{x}^H \mathbf{x} = 1$ 时，瑞利商退化为：$R(\mathbf{A}, \mathbf{x}) = \mathbf{x}^H \mathbf{A} \mathbf{x}$，这个形式在谱聚类和 PCA 中都有出现。

以上是瑞利商的内容，现在再看看广义瑞利商。广义瑞利商是指这样的函数 $R(\mathbf{A},\mathbf{B},\mathbf{x})$:

$$R(\mathbf{A},\mathbf{x})=\frac{\mathbf{x}^H\mathbf{A}\mathbf{x}}{\mathbf{x}^H\mathbf{B}\mathbf{x}} \tag{1.26}$$

其中 $\mathbf{x}$ 为非零向量，而 $\mathbf{A}$，$\mathbf{B}$ 为 $n\times n$ 的 Hermitan 矩阵。$\mathbf{B}$ 为正定矩阵。可以通过 Cholesky 分解（Cholesky decomposition）转化成标准瑞利商格式。令 $\mathbf{x}'=\mathbf{B}^{-1/2}\mathbf{x}$，则分母转化为：

$$\mathbf{x}^H\mathbf{B}\mathbf{x}=\mathbf{x}'^H(\mathbf{B}^{-\frac{1}{2}})^H\mathbf{B}\mathbf{B}^{-\frac{1}{2}}\mathbf{x}'=\mathbf{x}'\mathbf{H}\mathbf{B}^{-\frac{1}{2}}\mathbf{B}\mathbf{B}^{-\frac{1}{2}}\mathbf{x}'=\mathbf{x}'^H\mathbf{x}' \tag{1.27}$$

而分子转化为：

$$\mathbf{x}^H\mathbf{A}\mathbf{x}=\mathbf{x}'^H\mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}\mathbf{x}' \tag{1.28}$$

此时我们的 $R(\mathbf{A},\mathbf{B},\mathbf{x})$ 转化为 $R(\mathbf{A},\mathbf{B},\mathbf{x}')$:

$$R(\mathbf{A},\mathbf{B},\mathbf{x}')=\frac{\mathbf{x}'^H\mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}\mathbf{x}'}{\mathbf{x}'^H\mathbf{x}'} \tag{1.29}$$

利用前面的瑞利商的性质，我们可以很快的知道，$R(\mathbf{A},\mathbf{B},\mathbf{x})$ 的最大值为矩阵 $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$ 的最大特征值，或者说矩阵 $\mathbf{B}^{-1}\mathbf{A}$ 的最大特征值，而最小值为矩阵 $\mathbf{B}^{-1}\mathbf{A}$ 的最小特征值。

1.2.2. 二分类 LDA

回到 LDA 的问题上。假设我们的数据集 $D=(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,((\mathbf{x}_m,y_m))$，其中任意样本 $\mathbf{x}_i$ 为 n 维向量，$y_i\in\{0,1\}$。我们定义 $N_j(j=0,1)$ 为第 j 类样本的个数，$\mathbf{X}_j(j=0,1)$ 为第 j 类样本的集合，而 $\mu_j(j=0,1)$ 为第 j 类样本的均值向量，定义 $\sum_j(j=0,1)$ 为第 j 类样本的协方差矩阵（严格说是缺少分母部分的协方差矩阵）。

μ_j 的表达式为：

$$\mu_j=\frac{1}{N_j}\sum_{\mathbf{x}\in\mathbf{X}_j}\mathbf{x}\quad(j=0,1) \tag{1.30}$$

$\sum_j$ 的表达式为：

$$\sum_j=\sum_{\mathbf{x}\in\mathbf{X}_j}(\mathbf{x}-\mu_j)(\mathbf{x}-\mu_j)^T\quad(j=0,1) \tag{1.31}$$

由于是两类数据，因此只需要将数据投影到一条直线上即可。假设投影直线是向量 $\mathbf{w}$，则对任意一个样本 $\mathbf{x}_i$，它在直线 $\mathbf{w}$ 的投影为 $\mathbf{w}^T\mathbf{x}_i$，两个类别的中心点 μ_0 和 μ_1 在直线 $\mathbf{w}$ 的投影为 $\mathbf{w}^T\mu_0$ 和 $\mathbf{w}^T\mu_1$。由于 LDA 需要让不同类别数据的类别中心之间的距离尽可能的大，也就是要最大化 $||\mathbf{w}^T\mu_0-\mathbf{w}^T\mu_1||_2^2$；同时同一种类别数据的投影点尽可能的接近，也就是要同类样本投影点的协方差 $\mathbf{w}^T\sum_0\mathbf{w}$ 和 $\mathbf{w}^T\sum_1\mathbf{w}$ 尽可能的小，即最小化 $\mathbf{w}^T\sum_0\mathbf{w}+\mathbf{w}^T\sum_1\mathbf{w}$。综上所述，优化目标为最大化 $J(\mathbf{w})$：

$$J(\mathbf{w})=\frac{||\mathbf{w}^T\mu_0-\mathbf{w}^T\mu_1||_2^2}{\mathbf{w}^T\sum_0\mathbf{w}+\mathbf{w}^T\sum_1\mathbf{w}}=\frac{\mathbf{w}^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T\mathbf{w}}{\mathbf{w}^T(\sum_0+\sum_1)\mathbf{w}} \tag{1.32}$$

一般定义“类内散度矩阵”$\mathbf{S}_w$ 为：

$$\mathbf{S}_w=\sum_0+\sum_1=\sum_{\mathbf{x}\in\mathbf{X}_0}(\mathbf{x}-\mu_0)(\mathbf{x}-\mu_0)^T+\sum_{\mathbf{x}\in\mathbf{X}_1}(\mathbf{x}-\mu_1)(\mathbf{x}-\mu_1)^T \tag{1.33}$$

同时定义“类间散度矩阵”$\mathbf{S}_b$ 为：

$$\mathbf{S}_b=(\mu_0-\mu_1)(\mu_0-\mu_1)^T \tag{1.34}$$

这样优化目标重写为：

$$J(\mathbf{w})=\frac{\mathbf{w}^T\mathbf{S}_b\mathbf{w}}{\mathbf{w}^T\mathbf{S}_w\mathbf{w}} \tag{1.35}$$

这就是求解广义瑞利商极值。利用广义瑞利商的性质，知道 $J(\mathbf{w})$ 最大值为矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的最大特征值，而对应的 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的最大特征值对应的特征向量。

注意到对于二分类的时候，$\mathbf{S}_b\mathbf{w}$ 的方向恒为 $\mu_0-\mu_1$，不妨令 $\mathbf{S}_b\mathbf{w}=\lambda(\mu_0-\mu_1)$，将其带入：$(\mathbf{S}_w^{-1}\mathbf{S}_b)\mathbf{w}=\lambda\mathbf{w}$，可以得到 $\mathbf{w}=\mathbf{S}_w^{-1}(\mu_0-\mu_1)$，也就是说我们只要求出原始二类样本的均值和方差就可以确定最佳的投影方向 $\mathbf{w}$ 了。

1.2.3. 多类 LDA

类似地，对于多分类 LDA，假设我们的数据集 $D=(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,((\mathbf{x}_m,y_m))$，其中任意样本 $\mathbf{x}_i$ 为 n 维向量，$y_i\in\{C_1,C_2,\cdots,C_k\}$。我们定义 $N_j(j=1,2,\cdots,k)$ 为第 j 类样本的个数，$\mathbf{X}_j(j=1,2,\cdots,k)$ 为第 j 类样本的集合，而 $\mu_j(j=1,2,\cdots,k)$ 为第 j 类样本的均值向量，定义 $\sum_j(j=1,2,\cdots,k)$ 为第 j 类样本的协方差矩阵（严格说是缺少分母部分的协方差矩阵）。

由于多分类 LDA 是多类向低维投影，则此时投影到的低维空间不是一条直线，而是一个超平面。假设投影到的低维空间的维度为 d，对应的基向量为 $(\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_d)$，基向量组成的矩阵为 $\mathbf{W}_{n\times d}$。此时优化目标变成为:

$$J(\mathbf{W})=\frac{\mathbf{W}^T\mathbf{S}_b\mathbf{W}}{\mathbf{W}^T\mathbf{S}_w\mathbf{W}} \tag{1.36}$$

其中

$$\mathbf{S}_w=\sum_{j=1}^{k}\mathbf{S}_{wj}=\sum_{j=1}^{k}\sum_{\mathbf{x}\in\mathbf{X}_j}(\mathbf{x}-\mu_j)(\mathbf{x}-\mu_j)^T \tag{1.37}$$

$$\mathbf{S}_b=\sum_{j=1}^{k}N_j(\mu_j-\mu)(\mu_j-\mu)^T \tag{1.38}$$

$$\mu=\frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i \tag{1.39}$$

μ 为所有样本均值向量。

但是 $\mathbf{W}^T\mathbf{S}_b\mathbf{W}$ 和 $\mathbf{W}^T\mathbf{S}_w\mathbf{W}$ 都是矩阵，不是标量，无法作为一个标量函数来优化，我们无法直接用二类 LDA 的优化方法。一般来说，我们可以用其他的一些替代优化目标来实现。常见的一个 LDA 多类替代优化目标函数定义为：

$$J(\mathbf{W})=\frac{\prod_{diag}\mathbf{W}^T\mathbf{S}_b\mathbf{W}}{\prod_{diag}\mathbf{W}^T\mathbf{S}_w\mathbf{W}} \tag{1.40}$$

其中 $\prod_{diag}\mathbf{A}$ 为 $\mathbf{A}$ 的主对角线元素的乘积，这时 $J(\mathbf{W})$ 的优化过程可以转化为：

$$J(\mathbf{W})=\frac{\prod_{i=1}^{d}\mathbf{w}_i^T\mathbf{S}_b\mathbf{w}_i}{\prod_{i=1}^{d}\mathbf{w}_i^T\mathbf{S}_w\mathbf{w}_i}=\prod_{i=1}^{d}\frac{\mathbf{w}_i^T\mathbf{S}_b\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{S}_w\mathbf{w}_i} \tag{1.41}$$

这样就转化为了广义瑞利商。$J(\mathbf{W})$ 最大值是矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的最大特征值，最大的 d 个值的乘积就是矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的最大的 d 个特征值的乘积，此时对应的矩阵 $\mathbf{W}$ 为这最大的 d 个特征值对应的特征向量张成的矩阵。

由于 $\mathbf{W}$ 是一个利用了样本的类别得到的投影矩阵，因此它的降维到的维度 d 最大值为 $k-1$。因为 $\mathbf{S}_b$ 中每个 $\mu_j-\mu$ 的秩为 1，因此协方差矩阵相加后最大的秩为 k（矩阵的秩小于等于各个相加矩阵的秩的和），但是如果知道前 $k-1$ 个 μ_j 后，最后一个 μ_k 可以由前 $k-1$ 个 μ_j 线性表示，因此 $\mathbf{S}_jb$ 的秩最大为 $k-1$，即特征向量最多有 $k-1$ 个。

现在对 LDA 降维的流程做一个总结。

输入：数据集 $D=(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,((\mathbf{x}_m,y_m))$，其中任意样本 $\mathbf{x}_i$ 为 n 维向量，$y_i\in\{C_1,C_2,...,C_k\}$，降维到的维度 d

输出：降维后的样本集 D'

1. 计算类内散度矩阵 $\mathbf{S}_w$
2. 计算类间散度矩阵 $\mathbf{S}_b$
3. 计算矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$
4. 计算 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的最大的 d 个特征值和对应的 d 个特征向量 $(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d)$，得到投影矩阵 $\mathbf{W}$
5. 对样本集中的每一个样本特征 $\mathbf{x}_i$，转化为新的样本 $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$
6. 得到输出样本集 $D' = (\mathbf{z}_1, y_1), (\mathbf{z}_2, y_2), ..., ((\mathbf{z}_m, y_m))$

LDA 除了可以用于降维以外，还可以用于分类。一个常见的 LDA 分类基本思想是假设各个类别的样本数据符合高斯分布，这样利用 LDA 进行投影后，可以利用极大似然估计计算各个类别投影数据的均值和方差，进而得到该类别高斯分布的概率密度函数。当一个新的样本到来后，我们可以将它投影，然后将投影后的样本特征分别带入各个类别的高斯分布概率密度函数，计算它属于这个类别的概率，最大的概率对应的类别即为预测类别。

1.2.4. LDA 与 PCA 对比

LDA 用于降维，和 PCA 有很多相同，也有很多不同的地方，这里比较一下两者的降维异同点。

相同点：

1. 两者均可以对数据进行降维。
2. 两者在降维时均使用了矩阵特征分解的思想。
3. 两者都假设数据符合高斯分布。

不同点：

1. LDA 是有监督的降维方法，可以使用类别的先验知识经验，而 PCA 是无监督的降维方法，无法使用类别先验知识。
2. LDA 降维最多降到类别数 $k-1$ 的维数，而 PCA 没有这个限制。目前有一些 LDA 的进化版算法可以绕过这个问题。
3. LDA 选择样本点投影具有最大均值的方向，而 PCA 选择样本点投影具有最大方差的方向。换句话说，LDA 在样本分类信息依赖均值而不是方差的时候，比 PCA 较优；反之则亦然。即降维效果依赖于数据的均值和方差情况。
4. LDA 除了可以用于降维，还可以用于分类。

下图是数据分别在 LDA（d_2）与 PCA（d_1）下降维示意图：

1.3. 多分类问题和不平衡问题

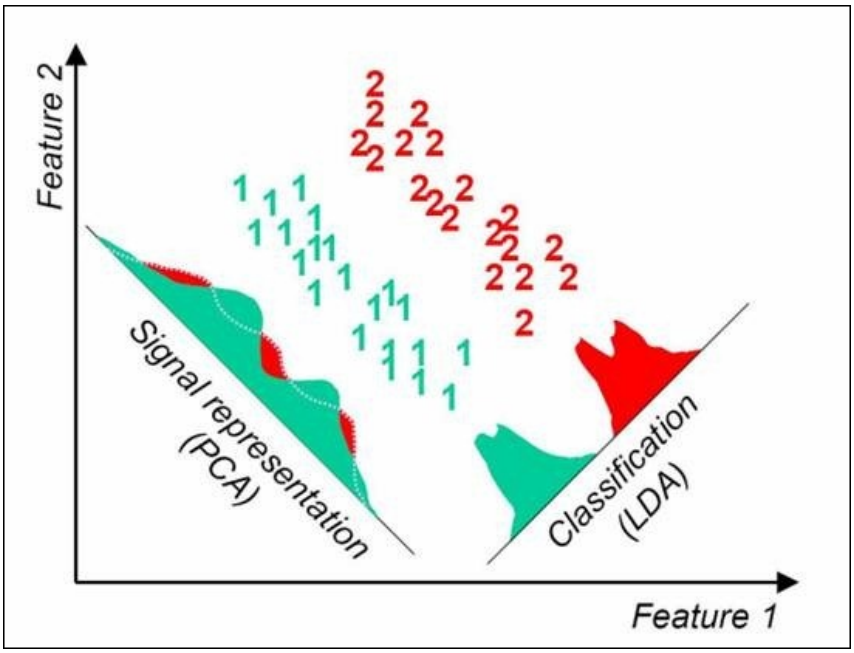

Figure 3. LDA 对比 PCA