机器学习 -- PCA, LDA和ICA

PCA, LDA 和ICA算法，常用于数据的降维分析，用于数据输入到其他模型前一些预处理，三者有相似之处，也各有不同。

1 统计学基础

给定一个m维随机变量$X = (x_1, x_2, ..., x_m)^T$，有n个观测样本构成了数据集$D$，记为$D = [X_1\ X_2\ ...\ X_n]$，观测样本$X_j = (x_{1j}, x_{2j}, ..., x_{nj})^T$, 则观测数据矩阵可用以下矩阵表示：

$$D = [X_1\ X_2\ ...\ X_n] = \begin{bmatrix} x_{11} & x_{12} & ... & x_{1n} \\ x_{21} & x_{22} & ... & x_{2n} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ x_{m1} & x_{m2} & ... & x_{mn} \end{bmatrix} \tag{1.1}$$

有以下定义：

1.1 散度矩阵

散度矩阵用于描述数据的离散程度，有以下定义

$$S = \sum_{j=1}^{n}(X_j - \bar{X})(X_j - \bar{X})^T \tag{1.2}$$

假设随机变量X共有M个类，记为$X = \{\Omega_1, \Omega_2, ..., \Omega_M\}$，共有n个样本记为$D = [X_1, X_2, ..., X_n]$,其中每类的样本数为$n_1, n_2, ..., n_M$，

- 类内散度矩阵

Ω_i类内的散度矩阵定义为：

$$S_w^{(i)} = \sum_{k=1}^{n_i}(X_k^{(i)} - \bar{X}^{(i)})(X_k^{(i)} - \bar{X}^{(i)})^T \tag{1.3}$$

总的类内散度矩阵为：

$$S_w = \sum_{i=1}^{M} S_w^{(i)} \tag{1.4}$$

从公式中我们可以观察到，类内散度描述的是某一类样本点与该类样本中心点的距离，总的类内散度矩阵刻画整体类内分布情况。

- 类间散度矩阵

Ω_i和Ω_j之间的散度矩阵（类内散度矩阵）定义为：

$$S_B^{ij} = (\bar{X}^{(i)} - \bar{X}^{(j)})(\bar{X}^{(i)} - \bar{X}^{(j)})^T \tag{1.5}$$

总的类间散度矩阵定义为：

$$S_B = \sum_{i=1}^{M} n_i(\bar{X}^{(i)} - \bar{X})(\bar{X}^{(i)} - \bar{X})^T \tag{1.6}$$

类间散度矩阵刻画的是两两类样本中心点的距离，两中心点靠得越近，则两类的距离越小，总的类间散度矩阵是整体样本的一个刻画。

- 总体散度矩阵

总体散度矩阵为：

$$S_T = S_B + S_w = \sum_{X_i \in D} (X_i - \bar{X})(X_i - \bar{X})^T \tag{1.7}$$

以上定义中，

第i类样本的均值为：

$$\bar{X}^{(i)} = \frac{\sum_{j=1}^{n_i} X_j^{(i)}}{n_i} \tag{1.8}$$

总体样本均值为：

$$\bar{X} = \frac{1}{n} \sum_{X_i \in D} X_i \tag{1.9}$$

2 PCA, LDA和ICA算法

2.1 PCA 主成分分析

主成分分析，也叫PCA(Principal Component Analysis), 假设原始数据有m维特征，则PCA的主要思想就是将m维特征映射到k维特征上， 且一般情况$m > k$,从m维到k维就完成了维度的压缩。这种映射我们可以理解维投影. 从m个维度中选择k个维度，我们自然而然就关注到这个k个维度要怎么选取，这也是PCA算法与接下来要提到的LDA算法的区别的地方.

2.1.1 PCA 算法的主要思想

在PCA算法中，将m维特征映射到k维特征，坐标轴选取的标准为： 选取的第一个坐标轴是原始数据方差最大的方向，选取的第二个坐标轴是与第一个坐标轴正交且方差最大的方向，依次类推，如下图，直线方向就是方差最大的方向，选取为第一主成分。

给出PCA算法的数学描述：

> 给定样本矩阵X, 样本的第一主成分$Y_1 = \alpha_1^T X$是在$\alpha_1^T \alpha_1 = 1$的条件下，使得$\alpha_1^T X_j (j = 1, 2, \dots, n)$的样本方差 $\alpha_1^T S \alpha_1$最大的X的线性变换， 样本第二主成分$Y_2 = \alpha_2^T X$是在$\alpha_2^T \alpha_2 = 1$和$\alpha_2^T X_j$与$\alpha_1^T X_j (j = 1, 2, \dots, n)$的样本协方差$\alpha_1^T S \alpha_2 = 0$条件下，使得$\alpha_2^T X_j (j = 1, 2, \dots, n)$的样本方差$\alpha_2^T S \alpha_2$最大的$X$的线性变换。 更一般地，样本的第$i$主成分$Y_i = \alpha_i^T X$是在$\alpha_i^T \alpha_i = 1$和$\alpha_i^T X_j$与$\alpha_k^T X_j (k < i, j = 1, 2, \dots, n)$的样本协方差$\alpha_k^T S \alpha_i = 0$条件下，使得$\alpha_i^T X_j (j = 1, 2, \dots, n)$的样本方差$\alpha_i^T S \alpha_i$最大的$X$的线性变换。

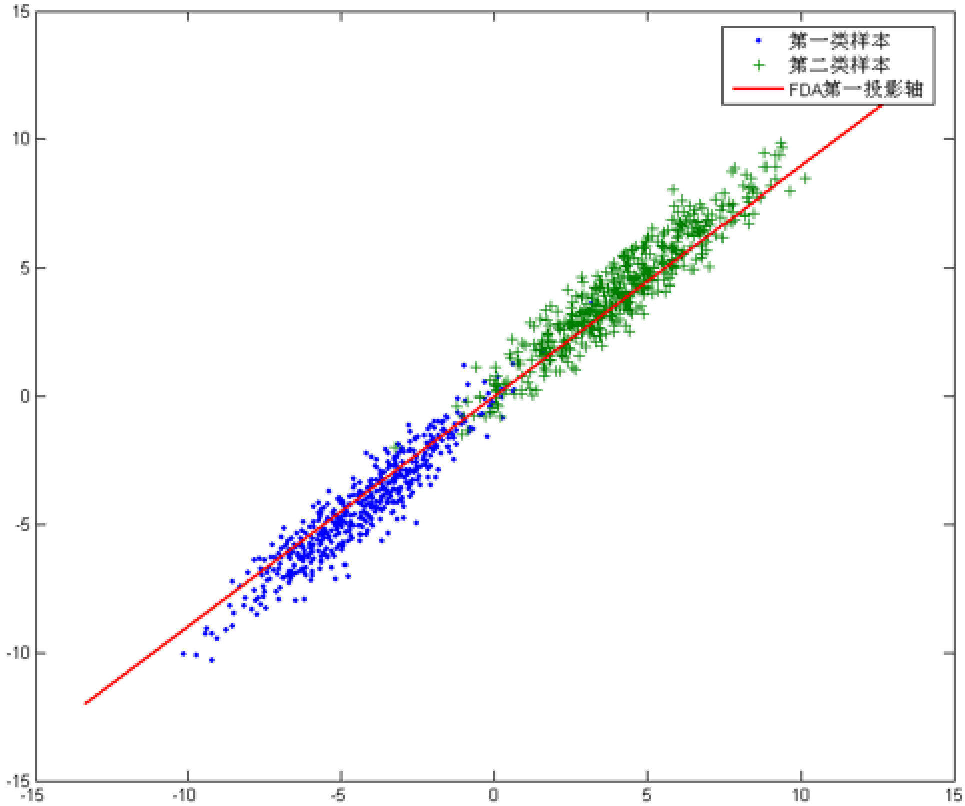
从原始的m维数据到变换后的k维数据，k的大小需要根据具体应用来确定，通常取k使得累计方差贡献率达到规定的百分比以上，累计方差贡献率反映了k个主成分保留的信息比例。

第k个主成分Y_k的方差贡献率定义为Y_k的方差与所有方差之和的比记作η_k，

$$\eta_k=\frac{\lambda_k}{\sum_{i=1}^{m}\lambda_i} \tag{2.1}$$

k个主成分$Y_1,Y_2,...,Y_k$的累计方差贡献率定义为k个方差之和与所有方差和之比

$$\sum_{i=1}^{k}\eta_i=\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \tag{2.2}$$



2.1.2 PCA 分解的两种方法

- 相关矩阵的特征值分解法 给定样本矩阵D，利用数据的协方差矩阵或者样本相关矩阵的特征值分解进行主成分分析，具体步骤如下：

(1) 对观测数据进行规范化处理， 得到规范化数据矩阵，记为X。

$$x_{ij}^* = \frac{x_{ij} - \bar{X}_i}{\sqrt{s_{ii}}}, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n \tag{2.3}$$

其中，$\bar{X}_i$是变量X第i维数据的均值

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij}, i = 1, 2, \ldots, m \tag{2.4}$$

s_{ii}是第i维数据的方差，

$$s_{ii} = \frac{1}{n-1} \sum_{j=1}^{n} (x_{ij} - \bar{X}_i)^2, i = 1, 2, \ldots, m \tag{2.5}$$

(2)对X计算相关矩阵R

$$R = [r_{ij}]_{m \times m} = \frac{XX^T}{n-1} \tag{2.6}$$

其中:

$$r_{ij} = \frac{\sum_{l=1}^{n} X_{il} X_{lj}}{n-1} \qquad i, j = 1, 2, \ldots, m \tag{2.7}$$

(3)求样本相关矩阵R的k个特征值和对应的k个单位特征向量。

$$|R - \lambda I| = 0 \tag{2.8}$$

得到R的m个特征值

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \tag{2.9}$$

求使方差贡献率达到预定值得主成分个数k, 方差贡献率计算公式如下:

$$\sum_{i=1}^{k} \eta_i \geq t \tag{2.10}$$

求前k个特征值对应得单位特征向量

$$\alpha_i = (\alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{mi})^T, i = 1, 2, \ldots, k \tag{2.11}$$

(4)求k个样本主成分，即以k个单位特征向量作为系数进行线性变换:

$$Y_i = \alpha_i^T X, \quad i = 1, 2, \ldots, k \tag{2.12}$$

(5) 计算k个主成分Y_j与原变量X_i的相关系数$\rho(X_i, Y_j)$，以及k个主成分对原变量X_i的贡献率υ_i

(6)计算n个样本的k个主成分值，将规范化样本数据代入到k个主成分式，得到n个样本的主成分值，$X_j = (x_{1j}, x_{2j}, \ldots, x_{mj})^T$的第$i$主成分值是

$$Y_{ij} = (\alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{mj})(x_{1j}, x_{2j}, \ldots, x_{mj})^T = \sum_{l=1}^{m} \alpha_{li} x_{lj}, i = 1, 2, \ldots, m; j = 1, 2, \ldots$$

- 数据矩阵的奇异值分解法

输入：$m \times n$的样本矩阵X, 其中每一行元素的均值为0；

输出：$k \times n$的样本主成分矩阵Y

参数：主成分个数k

(1) 构造新的$n \times m$矩阵

$$X' = \frac{1}{\sqrt{n-1}} X^T \tag{2.16}$$

X'的每一列均值为0.

(2) 对矩阵X'进行截断奇异值分解，得到

$$X' = U_k \Sigma_k V_k^T \tag{2.17}$$

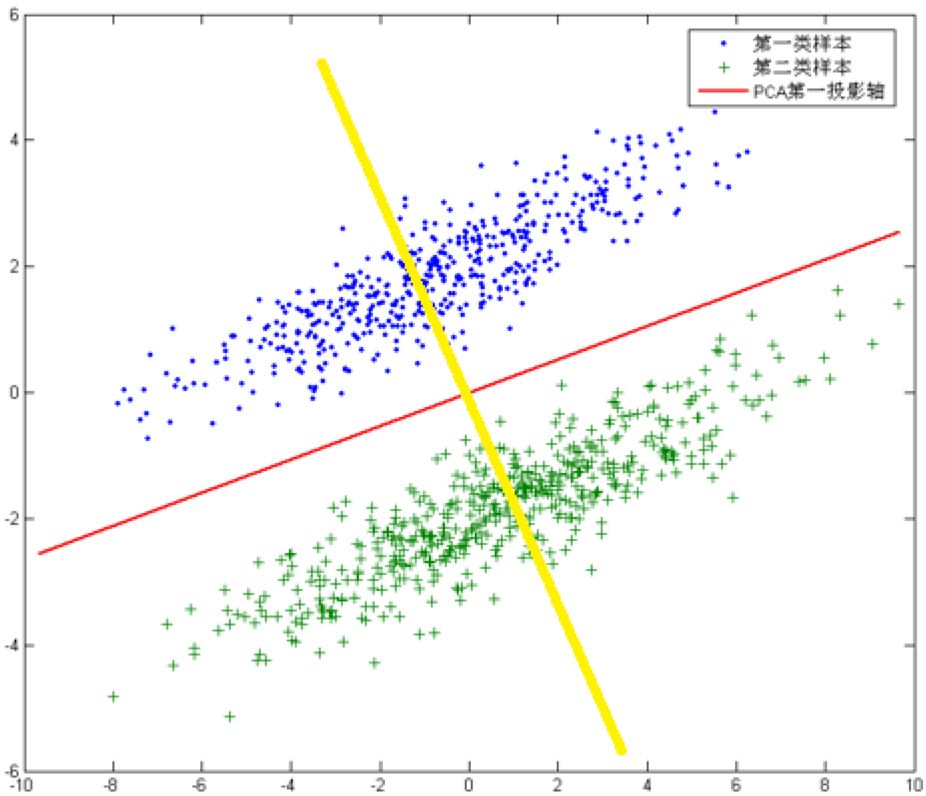
得到k个奇异值、奇异向量。矩阵V的前k列构成了k个样本主成分。

(3)求$k \times n$样本主成分矩阵

$$Y_k = V_k^T X \tag{2.18}$$

2.2 LDA 线性判别分析

与PCA相同LDA也是一种数据降维方法，两者都是将数据投影到新的相互正交的坐标轴上，但两者在投影过程中所使用的约束条件有所不同，上面我们提到PCA是将数据投影到方差最大且相互正交的坐标轴上，尽可能多的保留原有数据的信息。在类似上述那张图的数据分布中使用PCA算法是非常合适的，但是如果遇到以下情况(如图所示)，当我们把数据投影到方差最大方向的时候，将原本分离的数据给混合在了一起，增加了分析的困难。这时候PCA算法就不再适用，因此就引入了LDA算法。



通过观察我们可以发现，将数据投影到黄色直线上，既可以将数据分开，还可以压缩数据，而寻找黄线的过程便是LDA压缩数据的过程。

2.2.1 LDA算法的主要思想

2.2.1 LDA算法的主要思想

线性判别分析是一种监督学习方法，它将带标签的数据点通过投影的方法，投影到低维度中，使得样本点在低维度上可以更容易地被区分。这里的投影我们通常是指将向量投影到直线上， 一般地，对于样本矩阵$D \subset R^m$中的任意元素X_i,我们希望通过一个映射关系将其变换为R^k空间中的向量：

$$y = (y_1, y_2, \ldots, y_k)^T \tag{2.19}$$

其中$k < m$,因此这个映射可以完成数据降维的目的。已知X为一个n维列向量，给定一个n维列向量$\omega = (\omega_1, \omega_2, \ldots, \omega_m)^T$，通过

$$y = \omega^T X \tag{2.20}$$

就可以将X转化为一个数，若给定k个ω,

$$W_i = (\omega_{i1}, \omega_{i2}, \ldots, \omega_{im})^T, i = 1, 2, \ldots, k \tag{2.21}$$

引入矩阵

$$W = (W_1, W_2, \ldots, W_k) \tag{2.22}$$

通过映射关系

$$Y_i = W^T X_i \tag{2.23}$$

可以将n维向量X，映射成为k维向量

$$Y_i = (y_{i1}, y_{i2}, \ldots, y_{ik})^T, i = 1, 2, \ldots, m \tag{2.24}$$

其中X_i表示原始数据的任意一个样本值，Y_i表示映射后的样本值，由k个列向量组成的W矩阵又称为投影矩阵。LDA降维过程就是要求解出使投影后类内距离尽量小(属于同一类的样本尽可能地靠近)，类间距离尽量大（不同类样本尽量远）的投影矩阵W，达到该优化目标，就需要利用在第一部分提到的类间散度$S_B = \sum_{i=1}^{M} n_i(\bar{X}^{(i)} - \bar{X})(\bar{X}^{(i)} - \bar{X})^T$和类内散度$S_w = \sum_{i=1}^{M} S_w^{(i)}$。

上述S_B和S_w都是在原始R^m空间上定义的，这里我们直接给出将样本点投影到R^k维空间后的两个散度矩阵的结论

$$\hat{S_B} = W^T S_B W \tag{2.25}$$

$$\hat{S_w} = W^T S_w W \tag{2.26}$$

得到投影后的散度矩阵，我们定义优化目标函数J,

$$J(W) = \frac{det(\hat{S_B})}{det(\hat{S_w})} = \frac{W^T S_B W}{W^T S_w W} \tag{2.27}$$

我们知道，$\hat{S_B}$描述样本类间距离，$\hat{S_w}$描述样本类内距离，极大化目标函数$J(W)$我们只要极大化$det(\hat{S_B})$,极小化$det(\hat{S_w})$，这个优化与我们上面提到的优化目标是一样的。

不失一般性，令$W^T S_w W = 1$，则上式可以等价于：

$$min_W \quad W^T S_B W \tag{2.28}$$

$$st. \quad W^T S_w W \ = \ 1 \tag{2.29}$$

我们可以通过拉格朗日乘子法来优化上述目标函数,上式等价于：

$$C(W) = W^T S_B W - \lambda(W^T S_w W - 1) \tag{2.30}$$

对上式求导，最终可以的得到：

$$S_W^{-1} S_B W = \lambda W \tag{2.31}$$

这就变成了一个求解矩阵特征向量的问题。

2.2.2 LDA算法流程

使用LDA算法，对数据进行降维，具体步骤如下：

输入: 数据集$D \in R^m$,$D = \{(X_i, L_i)\}$,其中$i = 1, 2, \dots, n$表示一共有n个样本，$L_i \in \{1, 2, \dots, M\}$共有M类样本，L表示标签

输出：降维后的数据集,$D' \in R^k$

(1)根据公式（1.3），（1.4）计算类内散度矩阵S_w:

$$S_w = \sum_{i=1}^{M} S_w^{(i)} = \sum_{i=1}^{M} \sum_{k=1}^{n_i} (X_k^{(i)} - \bar{X}^{(i)})(X_k^{(i)} - \bar{X}^{(i)})^T$$

(2)根据公式（1.6）计算类间散度矩阵:

$$S_B = \sum_{i=1}^{M} n_i (\bar{X}^{(i)} - \bar{X})(\bar{X}^{(i)} - \bar{X})^T$$

(3)计算矩阵:

$$S_w^{-1} S_B$$

(4)对$S_w^{-1} S_B$进行奇异值分解，得到奇异值λ_i以及对应的特征向量ω_i,$i = 1, 2, \dots, n-1$

(5)取前k大的奇异值对应的特征向量组成投影矩阵W

(6)计算每个样本X_i在新的k维空间中的投影Y_i:

$$Y_i = W^T X_i$$

(7)得到降维以后的数据集D'

2.3 ICA 独立成分分析

独立主成分分析，又被称为ICA，是一种用来从多变量统计数据中寻找到隐含的因素或者成分的方法，被应用在多种领域，被认为是PCA和FA的一种拓展。ICA算法本质上是找出构成信号相互独立的部分，为观察数据定义了一个生成模型，在这个模型中，其认为观测数据矩阵X是由独立元（相互独立的部分）经过矩阵A线性加权获得，如下:

$$X = AS$$

2.3.1 ICA问题的表述

x为一个m维向量，$X \in R^m$,即

$$X = (x_1, x_2, \dots, x_m)^T \tag{2.32}$$

X的m个维度是相互非独立的，在一定假设条件下，我们可以用m个独立的变量的线性组合来重新表示X,如下:

$$(x_1, x_2, \dots, x_m)^T = A(s_1, s_2, \dots, s_m)^T \tag{2.33}$$

其中，s_i两两相互独立，A是满秩矩阵，$A \in R^{m \times m}$,令

$$S = (s_1, s_2, \dots, s_m)^T \tag{2.34}$$

则:

$$X = AS \tag{2.35}$$

又可以表示为:

$$S = WX \tag{2.36}$$

其中，$W = A^{-1}$,$W \in R^{m \times m}$ 假设X共有n个样本点，$D = \{X_1, X_2, \dots, X_n\}$,则，我们记数据矩阵如公式(1.1)的形式，独立主成分分析的目标就是在只知道数据矩阵的情况下，估算A, W, S的取值，其中最经典的实例：在一个大厅里，有m个人在聊天（分别对应着X的m个维度，两两之间不独立），在大厅的不同角落布置了m个麦克风记录大厅的声音，一共记录n秒（对应了n个样本点），ICA的目标就是从混合声音中把每个人的声音给单独分离出来。

2.3.2 ICA 问题求解

由前我们可知,

$$s_i = w_i X \tag{2.37}$$

其中，$w_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,m})$,每一个$w_i$都可以将x转化为一个数，即$S$的的一个维度$s_i$，设随机变量$s_i$概率密度函数是$p_{s_i}(s_i)$,同时我们假设x的概率密度函数为$p_x(x)$,我们是可以通过$s_i$的概率密度函数求x的概率密度函数的，这里直接给出结论:

$$p_X(x) = ||W|| \prod_{i=1}^{m} p_{s_i}(w_i X) \tag{2.38}$$

接下来要做的就是根据数据集D计算W的值，数据集D出现的概率是:

$$L = \prod_{i=1}^{n} (||W|| \prod_{j=1}^{m} p_{s_j}(w_j X_i)) \tag{2.39}$$

其中$X_i = (x_{1,i}, x_{2,i}, \ldots, x_{m,i})^T$，对(2.39)式两边取自然对数，则:

$$lnL = \sum_{i=1}^{n} (ln||W|| + \sum_{j=1}^{m} (lnp_{s_j}(w_j X_i))) = \sum_{i=1}^{n} \sum_{j=1}^{m} lnp_{s_j}(w_j X_i) + mln||W|| \tag{2.40}$$

当lnL取得最大值时，L也取得最大值，使用梯度下降法对（2.40）进行优化，最终可得W的更新公式为:

$$W = W + \alpha(Z^T D + n(W^{-1})^T) \tag{2.41}$$

其中α为学习率，Z为:

$$Z = g(K) = g(WD) \tag{2.42}$$

$$g(x) = \frac{1 - e^x}{1 + e^x} \tag{2.43}$$

理论上通过上述过程就可以迭代出数据集X出现时的W,进而求解出A,S,完成ICA问题的求解.

3 总结

（1）相比于其他两种算法，PCA算法具有更强的通用性。与ICA相比，PCA更容易获得稳定的主成分，同时与LDA相比较，PCA不要过分在意数据本身的类别信息，因为PCA算法为无监督学习算法。

（2）PCA、LDA的假设上都假设了数据的分布维高斯分布，而ICA是非高斯分布，因此在一些非高斯分布的数据集上，前两者的效果未必好。

（3）PCA与LDA在后面的几个步骤基本上是相同的，可以获得1~n维有排序关系的特征，一般情况下，排序越前的特征所包含的信息量越高。PCA旨在于去除原始数据中的冗余特征，是的投影在各个维度的方差尽可能大，而且是相互正交的。而LDA使得数据的类内距离小，类间距离大，较为关注的是样本的分类问题，其不保证投影到的新坐标系是正交的，也就是不同的数据之间可能还存在一定的相关性。与前者不同的是，ICA并不认为随机信号最有用的信息体现在类间或是最大方差里，而是构成样本集的独立成分，实际应用中ICA并不能起到降维的效果，也不单独使用，通常会和PCA或者白化处理结合使用。ICA的输出维数和输入相同，相互独立，没有排序关系，在某种意义上更具有区分度。其是一种数据预处理方式，在因果关系分析问题中应用广泛。

In [ ]: