PCA算法

总结一下PCA的算法步骤：

设有m条n维数据。

1）将原始数据按列组成n行m列矩阵X

2）将X的每一行（代表一个属性字段）进行零均值化，即减去这一行的均值

3）求出协方差矩阵$C = \frac{1}{m}XX^T$

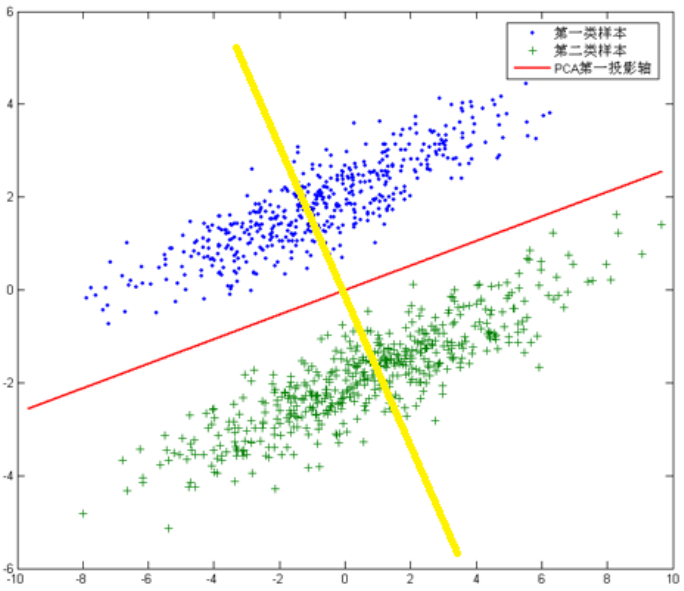
4）求出协方差矩阵的特征值及对应的特征向量

5）将特征向量按对应特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P

6）Y=PX即为降维到k维后的数据

PCA是一种无监督的数据降维方法，LDA是一种有监督的数据降维方法。即使训练样本有标签，PCA也不使用，但是LDA会利用数据的类别标签。

PCA和LDA都是将数据投影到新的相互正交的坐标轴上。只不过在投影的过程中他们使用的约束是不同的，也可以说目标是不同的。PCA是将数据投影到方差最大的几个相互正交的方向上，以期待保留最多的样本信息。样本的方差越大表示样本的多样性越好。如果样本信息不足，将导致模型性能不够理想。这就是PCA降维的目标：将数据投影到方差最大的几个相互正交的方向上。



上面这张图，红色是PCA方向，黄色是LDA方向。如果用PCA，数据将不再线性可分或者不可分。LDA降维的目标：将带有标签的数据降维，投影到低维空间同时满足三个条件：

- 尽可能多地保留数据样本信息（即选择最大的特征是对应的特征向量所代表的方向）
- 寻找使样本尽可能好分的最佳投影方向
- 投影后使得同类样本尽可能近，不同类样本尽可能远

符号

- x：表示训练样本，使用列向量表示
- x_i^j：表示第i类中的第j个样本
- C：表示有C类样本
- μ_i：表示第i类训练样本的均值 （i=1,2,...,C）
- M_i：表示第i类训练样本的数目
- M：表示训练样本的总数目 $M = \sum_{i=0}^{C} M_i$
- μ：是所有样本的均值向量 $\mu = \frac{1}{M}\sum_{i=1}^{M} x_i = \frac{1}{C}\sum_{i=1}^{C} \mu_i$
- D_i：表示第i类样本集合
- S_w：表示类内散度矩阵，w是within的简写
- S_b：表示类间散度矩阵，b是between的简写

优化目标

为什么是线性判别分析呢？所谓的线性就是，我们要将数据点投影到直线上（可能是多条直线），直线的函数解析式又称为线性函数。通常直线的表达式为

$$y = w^T x$$

其实这里的x就是样本向量（列向量），如果投影到一条直线上w就是一个特征向量（列向量形式）或者多个特征向量构成的矩阵。至于w为什么是特征向量，后面我们就能推导出来。y为投影后的样本点（列向量）。
我们首先使用两类样本来说明，然后再推广至多类问题。
将数据投影到直线w上，则两类样本的中心在直线上的投影分别为$w^T\mu_0$和$w^T\mu_1$，若将所有的样本点都都投影到直线上，则两类样本的协方差分别为$w^T\sum_0 w$和$w^T\sum_1 w$.
投影后同类样本协方差矩阵的求法：

$$\sum_{x\in D_i}(w^T x - w^T u_i)^2 = \sum_{x\in D_i}(w^T(x-u_i))^2 = \sum_{x\in D_i} w^T(x-\mu_i)(x-u_i)^T w = w^T\sum_{x\in D_i}[(x-\mu_i)(x-u_i)^T]w$$

上式的中间部分$\sum_{x\in D_i}(x-\mu_i)(x-u_i)^T$便是同类样本投影前的协方差矩阵。由还可以看出同类样本投影前后协方差矩阵之间的关系。如果投影前的协方差矩阵为Σ 则投影后的为 $w^T\Sigma w$.

上式的中间部分$\sum_{x\in D_i}(x-\mu_i)(x-u_i)^T$便是同类样本投影前的协方差矩阵。由还可以看出同类样本投影前后协方差矩阵之间的关系。如果投影前的协方差矩阵为Σ 则投影后的为 $w^T\Sigma w$.
上式的推导需要用到如下公式：a, b 都是列向量，$(a\cdot b)^2 = (a^T b)^2 = (a^T b)(a^T b) = (a^T b)(a^T b)T = a^T b b^T a$.
欲使同类样例的投影点尽可能接近，可以让同类样本点的协方差矩阵尽可能小，即$w^T\sum_0 w + w^T\sum_1 w$ 尽可能小; **而欲使异类样例的投影点尽可能远离，可以让类中心之间的距离尽可能大**，即$||w^T\mu_0 - w^T\mu_1||_2^2$ 尽可能大。同时考虑二者，则可得到欲最大化的目标

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

上式中的$||\cdot||$表示欧几里得范数，$||x-\mu_i||^2 = (x-\mu_i)^T(x-\mu_i)$

类间散度矩阵

类间散度矩阵其实就是协方差矩阵乘以样本数目，即散度矩阵与协方差矩阵只是相差一个系数。对于协方差矩阵和散度矩阵有疑问的同学可以参考博文：机器学习中的数学(3)——协方差矩阵和散布（散度）矩阵
对于两类样本而言：

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

对于多类问题，类间散度矩阵公式：

$$S_b = \sum_{i=1}^{C}(\mu_i - \mu)(\mu_i - \mu)^T$$

表示各个类样本均值的协方差矩阵。
如果我们只使用这样一个类间散度矩阵这样一个约束条件来对数据进行降维：即使得类间的样本投影后尽可能远离。那么参考PCA的降维过程：

$$S_b u = \lambda u$$

不同的是，为了保证类间的样本投影后尽可能远离，我们应该选择特征值最大的特征向量代表的方向做投影。这样才能保证，不同类样本投影之后方差尽可能地大，尽可能能地远离。

类内散度矩阵

对于两类问题而言：

$$S_w = \Sigma_0 + \Sigma_1 = \sum_{x \in D_0}(x - \mu_0)(x - u_0)^T + \sum_{x \in D_1}(x - \mu_1)(x - u_1)^T$$

对于多类问题类内散度矩阵公式：

$$S_w = \sum_{i=1}^{C}\sum_{j=1}^{M_i}(x_i^j - \mu_i)(x_i^j - \mu_1)^T$$

其中：

$$\sum_{j=1}^{M_i}(x_i^j - \mu_i)(x_i^j - \mu_i)^T$$

表示第i类样本的协方差矩阵。所以S_w就是表示C类样本协方差矩阵之和。
如果我们只使用这样一个类内散度矩阵这样一个约束条件来对数据进行降维：即使得类内的样本投影后尽可能接近。那么参考PCA的降维过程：

$$S_w u = \lambda u$$

不同的是，为了保证类内的样本投影后尽可能接近，我们应该选择特征值最小的特征向量代表的方向做投影。这样才能保证，同类样本投

优化

定义过类内散度矩阵和类间散度矩阵后，我们可以将上述的优化目标重新写为：

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

这就是LDA欲最大化的目标，即S_b与S_w的广义瑞利商。
如何确定w呢？注意到上式的分子和分母都是关于w的二次项，因此上式的解与w的长度无关,只与其方向有关.不失一般性，令$w^T S_w w = 1$,则上式等价于：

$$min_w - w^T S_b w$$

$$st. w^T S_w w = 1$$

使用拉格朗日乘子法,(对于拉格朗日乘子法不太了解的同学可以参考博文：机器学习中的数学(5)——拉格朗日乘子法和KKT条件)上式等价于：

$$c(w) = w^T S_b w - \lambda(w^T S_w w - 1)$$

$$\frac{dc}{dw} = 2S_b w - 2\lambda S_w w = 0$$

$$S_b w = \lambda S_w w$$

$$S_w^{-1} S_b w = \lambda w$$

可以看到上式就有转化为一个求解特征值和特征向量的问题了。w就是我们要求解的特征向量，这就验证了我们之前所说的式子$y = w^T x$中的w就是特征向量构成的矩阵。
但是到这里我们仍然有个问题需要解决，那就是S_w是否可逆。遗憾的是在实际的应用中，它通常是不可逆的，我们通常有连个办法解决该问题。

拓展

解决方法一：
令$S_w = S_w + \gamma I$, 其中γ是一个特别小的数，这样S_w一定可逆。
解决方法二：
先使用PCA对数据进行降维，使得在降维后的数据上S_w可逆，在使用LDA。