机器学习实验 7. 主成分分析 PCA

主成分分析算法原理概述

1. 主成分分析描述

主成分分析（PCA）是最重要的数据降维的方法之一。针对高维数据的处理时，往往会因为数据的高纬度产生大量的计算消耗，为了提高效率，一般最先想到方法就是对数据降维。与“属性子集选择”的方法（即选择一部分有代表意义的属性直接代替原数据）不同，PCA 是通过创建一个由原数据中的属性“组合”而成的，数量较小的变量集合代替原数据。所以 PCA 本质上是一种特征提取的方法，而非直接进行分类任务。

PCA 的基本思想可以这样描述：找出数据的所有属性中最主要的部分，用这个部分代替原始数据，从而达到降维的目的。显然，降维后的数据肯定会有所损失，而 PCA 的目的是要尽可能的保留原始数据的特征。所以，PCA 的核心在于如何寻找这个“最主要部分”。

比如，现在有一组二维数据集合，如图 Fig.1 所示，如果要对这些二维数据降维到一维，那很容易想到在这个坐标系中找到一条直线，然后将所有的二维数据点都映射到这条直线上，再处理这些映射后的点，就相当于是直接对一维数据做处理了。

- 哪个映射方向最好?

Fig.1 中，X 轴，Y 轴，l_1，l_2 四条线，到底映射到哪条线更好呢？显然是 l1，因为样本点投影在这条直线上能够尽可能的分开，可以理解为最大限

度的保留了数据的特征（反过来想，如果投影后，尽可能地分不开，那数据点不都一样了，也就没有了特征区分。）

同刚才所说，找超平面的理论依据是“在超平面上的投影点要尽可能的分开”，换句话说，就是要找到超平面，使之具备最大的“投影方差”。下面将进行详细的推导。

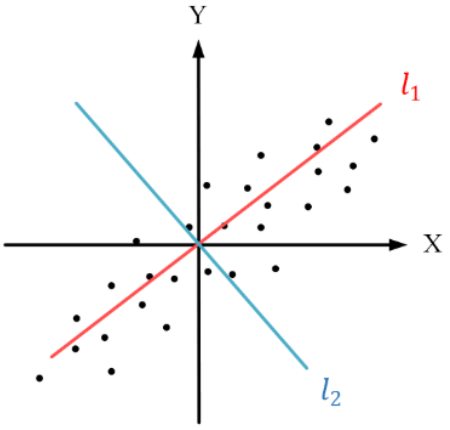

Fig. 1

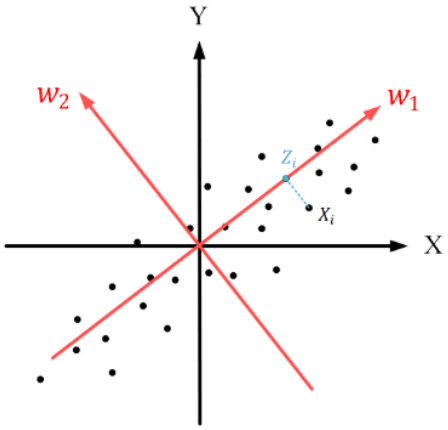

Fig. 2

主成分分析的求解与推导

对于 m 个 n 维数据向量$\{X_1, X_2, \ldots, X_m\}$，默认都已经经过了“中心化”处理，即$\sum_{i=1}^{m} Xi = 0$，如图 Fig.2 所示。

假设现在找到了最佳的超平面（维度是 k），则旋转现有的坐标系，使其中的 k 个坐标系组成的超平面就是要投影的超平面。这个过程相当于做基变换，变换过程中，令变换后的基是一组标准正交基，记为 $W=\{w_1,w_2,\ldots,w_n\}$。

$$W^T = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$

现在可以得到，在新的坐标系下，投影后数据元组的坐标表示是：

$$Z_i^* = (Z_{i1}, Z_{i2}, \cdots, Z_{in})^T = W^T X$$

知道了新坐标系下数据点的坐标，那么接着思考，这些数据点在其中一个或者多个新坐标轴构造的超平面上的投影坐标是不是可以得到了，实际上就是直接去除这一部分Z_i^*中的坐标即可，假设最终保留了 k 个坐标，可以得到下式：

$$Z_i = (Z_i^{(1)}, Z_i^{(2)}, \cdots, Z_i^{(k)})^T = W_0^T X_i$$

其中，W_0是正交基 W 去除了对应的列向量得到的，W_0^T 是由 k 个 n 维行向量构成的矩阵。至于是如何在Z_i^*中筛选出这 k 个坐标的，将在后面讨论。先假设我已经知道怎么筛选了。其中，我们得到了新坐标系下数据集的投影坐标。现在可以计算投影坐标的方差，然后进行优化计算，使得方差最大。因为 X_i（Z_i）已经经过了中心化处理，所以方差的计算公式可以写成如下形式：

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} Z_{ij}^2$$

注意，高维数据集的方差实际上是其每个维度上数据的方差和。我们最终优化的目标是要让方差最大。那么忽略常系数$\frac{1}{m}$，目标函数可以写成如下形式：

$$\max \sum_{i=1}^{m} Z_i^T Z_i$$

化简上式：

$$\sum_{i=1}^{m} Z_i^T Z_i \;=\; \sum_{i=1}^{m} X_i^T W_0 W_0^T X_i \;=\; tr(W_0^T X X^T W_0)$$

这样，可得到最终的优化问题：

$$\max \;\; tr(W_0^T X X^T W_0) \qquad \text{s.t. } W_0^T W_0 = E$$

其中，E 为 $k \times k$ 的单位矩阵。

用拉格朗日乘子法求解。先得到拉格朗日方程：

$$L(\lambda) = tr(W_0^T X X^T W_0) + \lambda(W_0^T W_0 - E)$$

对 W_0 进行求导，得到：

$$X X^T W_0 = -\lambda W_0$$

可见，W_0 为 XX^T 的特征向量组成的矩阵。因此，如果要将 n 维的数据降到 k 维，$tr(W_0^T X X^T W_0)$ 的取值就应该是矩阵 XX^T 的最大的 k 个特征值的和。

综上所述，如果我们事先不知道 k 的取值，可以先计算 XX^T 的特征值，选取其中明显较大的 k 个，并且以这 k 个特征值对应的特征向量组成的矩阵（记做 W_0）对原始的数据元组进行变换。

以上即为主成分分析原理的形象化解释。

主成分分析算法流程

1）预处理：对所有的数据进行中心化处理，$X_i = X_i - \frac{1}{m}\sum_{i=1}^{m} X_i$；

2）计算样本的协方差矩阵 XX^T，得到一个 n*n 的矩阵；

3）计算 XX^T 的特征值及特征向量，取特征值最大的 k 个特征向量做标准化处理，再作为列向量构成 W0（n*k 矩阵）；

4）对于每个数据 Xi，计算它在新坐标系 W0 中的坐标 $Z_i = X_i W_0$（Zi 为 k 维向量）

主成分分析算法的一般实现

1）导入模块：

```
from sklearn.decomposition import PCA
import numpy as np
```

2) 降维前的样本：

```
data = np.array([[0, 10, 2], [0, 10, 4], [0, 10, 6]])
print (data)
```

```
[[ 0 10  2]
 [ 0 10  4]
 [ 0 10  6]]
```

3) 降维后的样本：

```
newdata = pca.fit_transform(data)
print (newdata)
```

```
[[ 2.]
 [ 0.]
 [-2.]]
```

4) 投影方向向量：

```
print (pca.components_)
```

```
[[-0.  0. -1.]]
```

5) 导入数据：

```
iris = load_iris()

X = iris.data
y = iris.target
print (X.shape)
```

```
(150, 4)
```

```
iris_pca = PCA(n_components=2, copy=False, random_state=8)

X = iris_pca.fit_transform(X)
print (X.shape)
```

```
(150, 2)
```

6) 可视化结果：

```
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```