PCA

常见无监督学习任务

解决数据多重共线性问题，或者聚类

- **降维**：模型中变量越多，**多重共线性**和模型过拟合可能性越高。此外，大量的特征会增加模型的复杂性以及调参和拟合时间。减少数据集中的特征数量，即降维，将有助于消除多重共线性、提高泛化性能和训练速度。此外，还可用于数据可视化、数据压缩等。
- **聚类**：将高度相似的样本聚成组，使得组内高度同质性、组间高度异质性。可用于客户分组、搜索引擎、图像分割、半监督学习等。
- **异常检测**：学习正常数据，以便能从中检测出异常个例。可用于生产线上的次品检测等。
- **密度估计**：估计产生数据的随机过程的概率密度函数，密度估计结果常用于异常检测、数据分析、可视化。

有些数据集因为维数很高，会出现维数灾难的问题（样本稀疏，距离计算困难）

通过降维的方式，可以把高维特征空间转化为低维子空间。

降维(dimension reduction)

是缓解维数灾难的一个重要途径。通过某种**数学变换**，将原高维特征空间$\mathbb{R}^n$转变为一个低维“子空间”$\mathbb{R}^l\ (l < n)$，将x压缩在一个较小的表示中，同时损失的信息尽可能少。

主要途径两种：**投影**和**流形学习**。

PCA原理

PCA（主成分分析）的原理可以概括为一种**降维技术**，通过线性变换将高维数据映射到低维空间，同时尽可能保留原始数据的信息量（方差）。

1. **数据中心化**：对数据进行中心化处理，将每个特征的均值调整为0，目的是消除特征的偏置影响。
    - 数据中心化公式：$x' = x - \bar{x}$，其中 $\bar{x}$ 是特征均值。
2. **计算协方差矩阵**：通过数据矩阵计算各个特征之间的协方差，反映特征之间的相关性。
    - 协方差矩阵的公式：

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$$

3. **特征值分解**：对协方差矩阵进行特征值分解，得到特征值和特征向量。
    - 特征值表示数据在该方向上的方差大小。
    - 特征向量表示数据投影的方向。
4. **选择主成分**：根据特征值的大小选择前 k 个最大特征值对应的特征向量，作为新的坐标轴方向（主成分）。
    - 最大的特征值对应数据方差最大的方向。
5. **数据投影**：将原始数据投影到新的坐标轴上，实现降维。
    - 投影公式：

$$\mathbf{Z} = \mathbf{X}\mathbf{W}$$

其中 $\mathbf{W}$ 是选取的特征向量矩阵，$\mathbf{Z}$ 是降维后的数据。

找到高维数据中“方差最大”的方向，并将数据投影到这个方向上。

新的坐标轴（主成分）是相互正交的，能够捕捉数据中最主要的信息。

通过降维，可以去除冗余特征，简化数据分析和计算过程。

伪代码

输入： 数据集 $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(m)})^T$，$d$维空间，$m$个样本；
降维后空间的维数为k。

PCA对特征取值范围不同敏感

过程：

1. 标准化数据集，得$\boldsymbol{X}_{std}$；
2. 构造$\boldsymbol{X}_{std}$的协方差矩阵；
3. 计算协方差矩阵的特征值和特征向量；
4. 取最大的k个特征值所对应的特征向量$\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k$，构成投影矩阵$\boldsymbol{W}$。
5. 用投影矩阵$\boldsymbol{W}$对d维输入数据集$\boldsymbol{X}$进行变换，获得新的k维特征子空间投影 $\boldsymbol{X}_{pca} = \boldsymbol{X} \cdot \boldsymbol{W}$。

输出： 投影矩阵$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_k)$，降维后的数据集$\boldsymbol{X}_{pca}$。

1：标准化数据集，得$\boldsymbol{X}_{std}$
2：对$\boldsymbol{X}_{std}$做SVD分解
3：取前k个右奇异向量构成投影矩阵$\boldsymbol{W}$

```
def pca(X,topN):
"""参数：原始数据X（数组），低维空间维度topN（整数）"""
  sc = StandardScaler()
  X_std = sc.fit_transform(X) # 1.标准化数据
  U,S,Vt = np.linalg.svd(X_std) # 2.调SVD()分解函数，返回U,s, V ：X_std的右奇异矩阵的转置
  W = Vt.T[:,:topN] # 3.取前topN个右奇异向量构成投影矩阵
  X_pca = X_std.dot(W) # 4.将原始数据转换到新空间，得到降维后数据
  X_recon = sc.inverse_transform(X_pca * W.T) # 5.降维后数据重构，用于与原始数据比较
  return X_pca,X_recon
```

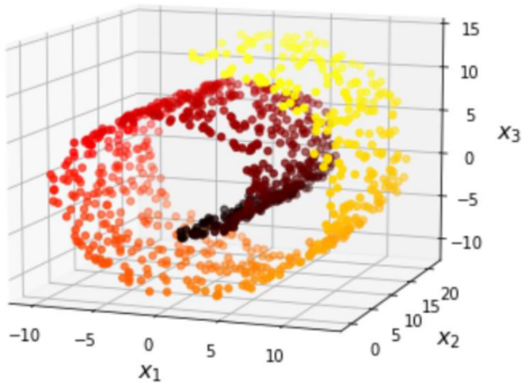
PCA算法的分类

- **随机PCA算法**　随机选出主成分，速度快
- **增量PCA算法**　小批处理，解决内存受限问题

  from sklearn.decomposition import IncrementalPCA
- **核PCA算法**　核技巧，实现复杂的非线性投影

  from sklearn.decomposition import KernelPCA　非线性降维

流形学习



大多数真实世界中的高维数据集都接近于低维流形(manifold)。—— **流形假设**

左图，瑞士卷是一个2维流形，一个在3维空间中被卷曲的2维形状。

流形是指嵌在高维空间中的低维拓扑空间。一个d维流形是一个n维空间(d<n)中的一部分，它局部地类似于一个d维的超平面，在n维空间中被弯曲、扭曲。

许多降维算法通过对训练数据所在的流形建模来工作，这称为**流形学习（manifold learning）**。

流形假设还伴随另一个**隐式假设**：低维流形对于后续的分类或回归任务更简单。

LLE：流形学习算法，主要用于非线性降维