Abstract

本期我们开始学习降维的算法。

我们知道，解决过拟合的问题除了增加数据和正则化之外，降维是最好的方法。

实际上，早先前辈们就遇见过维度灾难。我们知道 n 维球体的体积为

$$CR^n \tag{9}$$

因此球体的体积与 n 维超立方体的比值为

$$\lim_{n\rightarrow+\infty} = \frac{CR^n}{2^nR^n} = 0$$

由公式我们可以看出，在高维数据中，样本的分布是相当稀疏的，超立方体的内部基本上是空心的，因此对数据的建模增大了难度。这就是所谓的维度灾难。

降维的方法分为：

- 直接降维，特征选择
- 线性降维，PCA，MDS等
- 分线性，流形包括 $Isomap$，LLE 等

Idea

对于PCA的核心思想，老师总结了一句顺口溜：一个中心，两个基本点

- 一个中心：
  - 将原本可能线性相关的各个特征，通过正交变换，变换为一组线性无关的特征
  - 即对原始特征空间的重构。
- 两个基本点：
  - 最大投影方差
    - 使数据在重构后的特征空间中更加分散(因为原始的数据都是聚为一堆分散在角落的)
  - 最小重构距离
    - 使得数据在重构之后，损失的信息最少(即在补空间的分量更少)

Algorithm

下面我们主要讲述第一个基本点：最大投影方差，其实两个基本点都是一个意思，只不过是从不同的角度对一个中心进行诠释。

首先是投影，关于投影的知识，我们前面已经讲过了，这里也是一样。我们假设样本点x_i，一个基向量u_i，假设$u_i^Tu_i=1$，因此可以得到样本在u_i这个维度的投影为

$$project_i = x_i^Tu_i \tag{10}$$

而样本经正交变换后原本有p个特征维度，因我们需对其降维，因此只取其前q个特征，而这q个特征都是线性无关的，因此可以将这些投影直接相加，得到样本在新的特征空间的投影。

注意在求投影之前先将数据做中心化，因此数据的均值归零，求投影的方差可以直接平方。

综上，我们得到了目标函数：

$$J=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{q}\left((x_i-\bar{x})^T u_j\right)^2 \tag{11}$$

下面对目标函数稍作推导：

因为$((x_i-\bar{x})^T u_j)$的形状为$(1,p)*(p,1)=(1,1)$，因此可以对其做转置：

$$\begin{aligned}
J&=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{q}\left((x_i-\bar{x})^T u_j\right)^2\\
&=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{q}(u_j^T(x_i-\bar{x}))^2\\
&=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{q}u_j^T(x_i-\bar{x}))(x_i-\bar{x})^T u_j\\
&=\sum_{j=1}^{q}u_j^T(\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x}))(x_i-\bar{x})^T)u_j\\
&=\sum_{j=1}^{q}u_j^T S u_j
\end{aligned} \tag{12}$$

别忘了我们还有一个限制条件：$s.t\ u_j^T u_j=1$

因此可以使用拉格朗日乘子法:
$\underset{u_j}{\operatorname{argmax}} L\left(u_j,\lambda\right)=\underset{u_j}{\operatorname{argmax}} u_j^T S u_j+\lambda\left(1-u_j^T u_j\right)$
对上式求导:

$$\frac{\partial\Delta}{\partial u_j}=2Su_j-2\lambda u_j=0 \tag{13}$$

得到结果:

$$Su_j=\lambda u_j \tag{14}$$

可以看出，变换后的基向量实际上为协方差矩阵的特征向量，λ为S的特征值

实际上，对于协方差矩阵的求解也可以化简：

$$\begin{aligned}
S&=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})(x_i-\bar{x})^T\\
&=\frac{1}{N}(x_1-\bar{x},x_2-\bar{x},\cdots,x_N-\bar{x})(x_1-\bar{x},x_2-\bar{x},\cdots,x_N-\bar{x})^T\\
&=\frac{1}{N}\left(X^T-\frac{1}{N}X^T I_N I_N^T\right)\left(X^T-\frac{1}{N}X^T I_N I_N^T\right)^T\\
&=\frac{1}{N}X^T\left(E_N-\frac{1}{N}I_N I_N^T\right)\left(E_N-\frac{1}{N}I_N I_N^T\right)^T X\\
&=\frac{1}{N}X^T H_N H_N^T X\\
&=\frac{1}{N}X^T H_N H_N X=\frac{1}{N}X^T H X
\end{aligned} \tag{15}$$

这里H是一个特殊的矩阵，被称为中心矩阵。

$$H = E_N - \frac{1}{N} I_N I_N^T \tag{16}$$

因此，在实作中，我们只需要用上式求出协方差矩阵，然后对其做正交分解得到特征值与特征向量即可。

Implement

```
import numpy as np
import os
os.chdir("../")
from models.decompose_models import PCA

k, b = 3, 4
x = np.linspace(0, 10, 100)
y = x * k + b
x += np.random.normal(scale=0.3, size=x.shape)
data = np.c_[x, y]

model = PCA()
model.fit(data)
model.draw(data)
```