PCA 主成分分析

> 2018.10.02 Louis

在高维特征空间中，数据样本稀疏，且大多数点都分布在边界处、实例彼此远离，预测的可靠性更低。训练集维度越高，过拟合风险越大。

降维的方法主要有“投影”和“流形学习”。

PCA 是一种投影的降维方法。

---

1. 西瓜书的 PCA 解释

Principal Component Analysis 主成分分析是一种常见的降维方法。首先它找到接近数据集分布的超平面，然后将所有的数据都投影到这个超平面上。

这个超平面要具备怎样的性质?

- 最近重构性：样本点到超平面距离足够近
- 最大可分性：样本点到超平面的投影尽可能分开

实际上，超平面的这两种性质是等价的。

这张图能直观地解释

1.1 最近重构性

PCA 假定数据集以原点为中心，首先样本数据要进行中心化，即 $\sum_i \boldsymbol{x}_i = \mathbf{0}$，投影变换后得到的坐标系为$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$，每个 w_i 都是标准正交基向量，$\|\boldsymbol{w}_i\|_2 = 1, \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0 (i \neq j)$。

再丢弃新坐标系中的部分坐标，维度从 d 维降低为 d' 维，$d' < d$，则原来样本点 x_i 在 d' 维坐标系中的投影是 $\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$，其中 $z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$ 是 x_i 在低维坐标系下第 j 维的坐标。

若基于 z_i (即投影后的样本点坐标)，来重新构造出原来坐标系下的样本点 x_i，则会得到 $\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j$.

再来考虑整个训练集中，原样本点 x_i 与基于投影重构的样本点 $\hat{\boldsymbol{x}}_i$ 直接的距离：

$$\begin{aligned}\sum_{i=1}^{m} \left\| \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j - \boldsymbol{x}_i \right\|_2^2 &= \sum_{i=1}^{m} z_i^{\mathrm{T}} \boldsymbol{z}_i - 2 \sum_{i=1}^{m} \boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \boldsymbol{x}_i + \text{ const} \\ &\propto -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}} \left(\sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \right) \mathbf{W} \right)\end{aligned}$$

> 我还没想明白为什么正比于那个迹

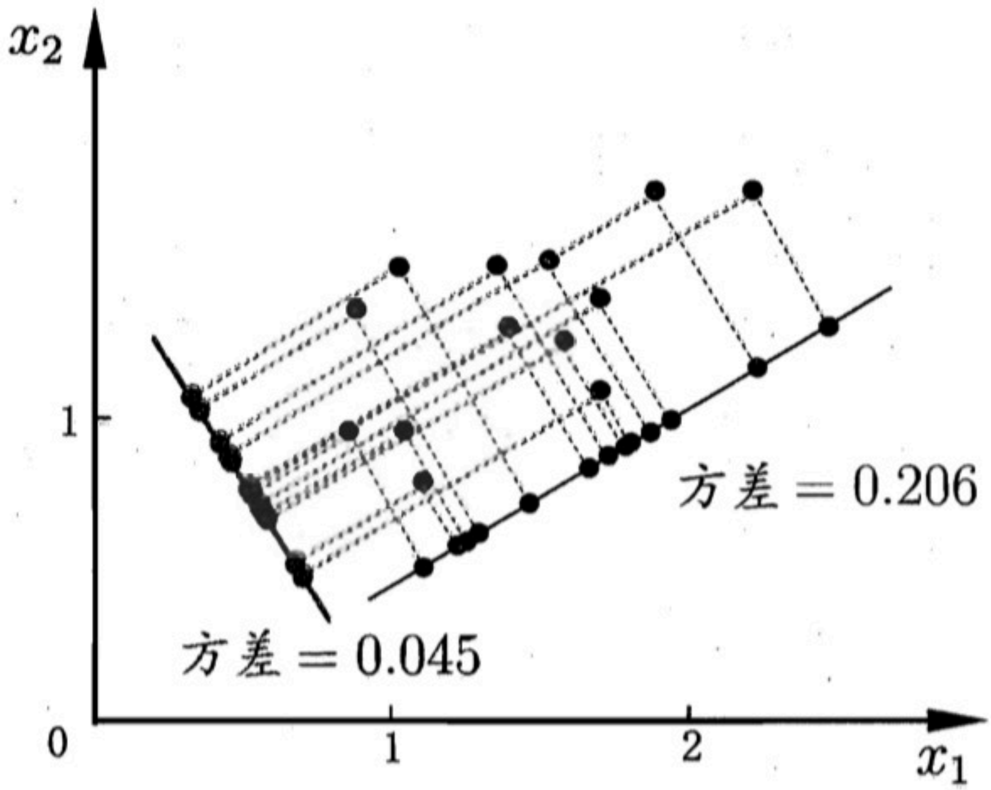
这样，**PCA 的优化目标为**

$$\begin{aligned}&\min_{\mathbf{W}} -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}} \mathbf{X} \mathbf{X}^{\mathrm{T}} \mathbf{W}\right) \\ &\text{s.t. } \mathbf{W}^{\mathrm{T}} \mathbf{W} = \mathbf{I}\end{aligned}$$

w_j 是标准正交基，$\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵（实际上乘 1/(m-1) 才是，但常数项不影响）

1.2 最大可分性

样本点 x_i 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}} \boldsymbol{x}_i$，若所有样本点投影尽可能分开，则应该使得投影后样本点的方差最大化：如图所示



投影后样本点的方差是 $\sum_i \mathbf{W}^{\mathrm{T}} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \mathbf{W}$，于是优化目标写为：

$$\begin{gathered}\max_{\mathbf{W}} \operatorname{tr}\left(\mathbf{W}^{\mathrm{T}} \mathbf{X} \mathbf{X}^{\mathrm{T}} \mathbf{W}\right) \\ \text{s.t. } \mathbf{W}^{\mathrm{T}} \mathbf{W}=\mathbf{I}\end{gathered}$$

显然，这与最近重构性的优化目标等价。

对上面两式使用拉格朗日乘子法得到

$$\mathbf{X} \mathbf{X}^{\mathrm{T}} \mathbf{W}=\lambda \mathbf{W}$$

只需要对协方差矩阵 $\mathbf{X} \mathbf{X}^{\mathrm{T}}$ 进行特征值分解，将求得的特征值排序：$\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_d$，再取前 $d^{\prime}$ 个特征值对应的特征向量构成 $\mathbf{W}=\left(w_1, w_2, \ldots, w_{d^{\prime}}\right)$，这就是主成分分析的解。

1.3 算法

输入: 样本集 $D=\left\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\right\}$;

过程:

1. 对所有样本进行中心化：$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i-\frac{1}{m} \sum_{i=1}^m \boldsymbol{x}_i$
2. 计算样本的协方差矩阵 $\mathbf{X} \mathbf{X}^{\mathrm{T}}$
3. 对协方差矩阵 $\mathbf{X} \mathbf{X}^{\mathrm{T}}$ 进行特征值分解
4. 取最大的 $d^{\prime}$ 个特征值所对应的特征向量$\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d^{\prime}}$

输出: 投影矩阵 $\mathbf{W}=\left(w_1, w_2, \ldots, w_{d^{\prime}}\right)$

> 实践中，通常对 X 进行奇异值分解来代替协方差矩阵的特征值分解。

降维后低维空间的维数 $d^{\prime}$ 通常由用户事先指定，或通过在 $d^{\prime}$ 值不同的低维空间中对 k 近邻分类器（或其他开销较小的的学习器）进行**交叉验证**来选取较好的 $d^{\prime}$ 值，对 PCA，还可以从重构的角度设置一个重构阈值， 例如 $t=95\%$，然后选取使下式成立的最小 $d^{\prime}$ 值.

$$\frac{\sum_{i=1}^{d^{\prime}} \lambda_i}{\sum_{i=1}^d \lambda_i} \geqslant t$$

降维舍弃部分信息是必要的：

1. 舍弃部分信息后能使采样密度增大，这是降维的重要动机
2. 最小特征值所对应的特征向量往往与噪声有关，舍弃它们一定程度上有去噪的效果

---

2. 其他解释

2.1 推导

西瓜书上的部分推导较简略，还看到另一个比较详细的推导。

现在只考虑**最小化投影造成的损失**，即最近重构性。

现在标准正交基设为 $\{u_j\}, (j=1,\ldots,n)$，我们**要减掉其中一些维度**，使得误差足够小。

对 $\mathbf{x_i}$ 在方向 $\mathbf{u_j}$ 上的投影是 $(x_i^T u_j)u_j$

如果减掉 $\mathbf{u_j}$ 这个维度，造成的误差为：

所有样本在 u_j 维度上的投影的平方和取均值。

$$\begin{aligned} J_j &= \frac{1}{m}\sum_{i=1}^{m}\left(x_i^T u_j\right)^2 \\ &= \frac{1}{m}\left\|x^T u_j\right\|_2 \\ &= \frac{1}{m}\left(x^T u_j\right)^T\left(x^T u_j\right) \\ &= \frac{1}{m}u_j^T x x^T u_j \end{aligned}$$

> 注意如何将$m\times 1$ 维度的 $L2$ 范式转换为 $v^T v$ 的技巧，其中 v^T 维度为 $1\times m$，而 v 维度为 $m\times 1$，其实就是每个元素的平方和。

将 $\frac{1}{m}xx^T$ 记作 S，接下来要考虑我们要减去哪 t 个维度，使得损失最小化：

$$J=\sum_{j=n-t}^{n}u_j^T S u_j$$

$$s.t.\quad u_j^T u_j = 1$$

此时使用**拉格朗日乘子法**使得：

$$\tilde{J}=\sum_{j=n-t}^{n}u_j^T S u_j+\lambda_j\left(1-u_j^T u_j\right)$$

最小化上式子，求导有：

$$\frac{\delta\tilde{J}}{\delta u_j}=Su_j-\lambda_j u_j$$

> 上面是标量对矢量求导，注意求导后的维度与被求导的维度(u_j)要是一致的。

使其为 0 则得到：

$$Su_j=\lambda_j u_j$$

这是标准的特征值的定义，λ_j 是特征值，u_j 是对应的特征向量，所以对 S 进行特征值分解即可得到解，将上式代入原始的 J 中，得到：

$$
\begin{aligned}
J &= \sum_{j=n-t}^{n} u_j^T S u_j \\
&= \sum_{j=n-t}^{n} u_j^T \lambda_j u_j \\
&= \sum_{j=n-t}^{n} \lambda_j
\end{aligned}
$$

所以要使 J 最小，就去掉变换后维度中最小的 t 个特征值对应的维度就好了。

2.2 疑问展开

一、为什么 $\frac{1}{m-1}xx^T$ 就是协方差矩阵呢?

对于两个变量 X,Y，其协方差 $Cov(X,Y)$为

$Cov(X,Y)=E[X-E(X)][Y-E(Y)]$

现在是对向量而言，协方差矩阵的计算公式为：

$$\Sigma=\mathrm{E}\left[(\mathbf{x}-\mathrm{E}[\mathbf{x}])(\mathbf{x}-\mathrm{E}[\mathbf{x}])^{\top}\right]$$

由于我们第一步进行了去中心化 $x-E[x]$，所以 $s=\frac{1}{m}xx^T$其实就是协方差矩阵（虽然标准的协方差矩阵上应该是 1/(m-1)，但系数对特征值、特征向量无影响）

2.3 numpy 代码

> 下面代码中的矩阵 X 其实是上面推导中的 X^T，每一行是一个样本

```
def pca(X):
    # Step1: 去中心化
    mean_ = np.mean(X, axis=0)
    X = X - mean_

    # Step2: 得到协方差矩阵
    covMat = np.cov(X,rowvar = 0) # rowvar=False 每列代表一个变量
    # 实际上是否去中心化对于得到协方差矩阵无影响，  只是为了方便后续降维

    # Step3: 得到特征值和特征向量 eigenvalue, eigenvector
    eigVal,eigVec = sp.linalg.eig(covMat)

# 不用 cov，只用矩阵乘法
def pca_dot(X):
    mean_ = np.mean(X, axis=0)
    X = X - mean_
    M,N=X.shape
    Sigma=np.dot(X.transpose(),X)/(M-1)
    eigVal,eigVec = sp.linalg.eig(Sigma)
```

2.4 PCA 和 SVD 的关系

简而言之，就是 `SVD` 奇(qí)异值分解，在 PCA 的应用中常用来代替特征值分解。

用特征值分解，矩阵中一些较小的数容易在平方中丢失。而 `SVD` 分解不直接计算 X^TX，所以不会丢失较小的数，而且速度比特征值分解快很多，充分利用了协方差矩阵的性质。

> 奇异值分解 （Wikipedia） $X=U\Sigma V^*$ U 是 $m\times m$ 阶 `酉矩阵`，
>
> Σ (sigma) 是 $m\times n$ 阶**非负实数对角矩阵**
>
> 而V^*，即 V 的**共轭转置**，是 $n\times n$ 阶 `酉矩阵`
>
> 这样的分解就称作 X 的**奇异值分解**
>
> 1. U 的 **列** 组成一套对 X 的正交"**输出**"的基向量，这些向量是XX^T 的**特征向量**。
> 2. Σ **对角线**上的元素是**奇异值**，可视为是在输入与输出间进行的标量的"**膨胀控制**"。这些是 X^TX 和 XX^T **特征值**的非零平方根，并与U 和 V 的**行向量**相对应。
> 3. V 的 **列** 组成一套对 X 的正交"**输入**"或"**分析**"的基向量。这些向量是 X^TX 的**特征向量**。

因此 `SVD` 的结果得到的特征向量，可以直接用于 PCA 降维。

```
# 求出协方差矩阵
def pca_svd_cov(X):
    mean_ = np.mean(X, axis=0)
    X = X - mean_
    M,N=X.shape
    Sigma=np.dot(X.transpose(),X) #这里直接去掉/(M-1)方便和pca_svd比较，对求得特征向量无影响
    U,S,V = sp.linalg.svd(Sigma); #  把 eig 改成 svd
    eigVal,eigVec = S,U
```

结论一：协方差矩阵（或 X^TX ）的奇异值分解结果和特征值分解结果一致

```
# 不求协方差矩阵，通过 SVD 也可以直接得出 X^T X 的特征向量
def pca_svd(X):
    mean_ = np.mean(X, axis=0)
    X = X - mean_
    U, S, V = sp.linalg.svd(X)
    # S 对角线就是特征值非零平方根•, V 列向量就是特征值
    eigVal,eigVec = S,V
```

结论二：V 的**列**组成一套对 X 的正交"输入"或"分析"的基向量。这些向量是 X^TX的特征向量。

2.4.1 结论二 的推导

根据奇异值分解的定义：

$$X=U\Sigma V^T$$

则有

$$
\begin{aligned}
X^TX &= V\Sigma U^TU\Sigma V^T \\
&= V\Sigma^2V^T \\
&= V\Sigma^2V^{-1}
\end{aligned}
$$

Σ 是对角矩阵，U 是标准正交基（酉矩阵），V 是标准正交基 （$VV^T = I; V = V^{-1}$）

X^TX 是一个对称的半正定矩阵，它可以通过特征值分解为

$$X^TX = Q\Lambda Q^{-1}$$

（Λ 是对角化特征值,Q 是特征向量）

可以看到 $X^TX = V\Sigma^2V^{-1}$ 和 $X^TX = Q\Lambda Q^{-1}$ 形式一致，当限定了特征值顺序后，这样的组合是唯一的，所以**结论二**成立：V 是 X^TX 的特征向量，奇异值和特征值是平方关系：

$$
\begin{aligned}
V &= Q \\
\Lambda &= \Sigma^2
\end{aligned}
$$

所以 `u, s, v` 得到的奇异值 `s` 的平方才是特征值 `eigval`，可以通过运行代码得到验证

2.4.2 结论一 的推导

结论一：协方差矩阵（或 X^TX ）的奇异值分解结果和特征值分解结果一致

我们对 X^TX 进行 SVD 分解，为了区分，U 取下标 2

$$X^TX = U_2\Sigma_2V_2^T$$

注意是：

$$X = U\Sigma V^T$$

SVD 分解性质的第二条：

> U 的列组成一套对 X 的正交"输出"的基向量，这些向量是 XX^T的特征向量
>
> 注意这里 X 是 X^TX，所以 U_2 的列是矩阵 $X^TXX^TX = (X^TX) * (X^TX)^T$ 的特征向量

$$
\begin{aligned}
X^TXX^TX &= U_2\Sigma_2V_2^T\left(U_2\Sigma_2V_2^T\right)^T \\
&= U_2\Sigma_2^2U_2^T
\end{aligned}
$$

$$
\begin{aligned}
X^TX &= Q_2\Lambda_2Q_2^{-1} \\
X^TXX^TX &= Q_2\Lambda_2^2Q_2^{-1}
\end{aligned}
$$

所以有：

$$
\begin{aligned}
U_2 &= Q_2 \\
\Sigma^2 &= \Lambda^2
\end{aligned}
$$

能得到这样的结果是因为 $X_T X$ 本身是对称的半正定矩阵。

2.5 其他

`SVD` 还常用来计算伪逆，这在最小二乘中有应用：

$$X^{-1} = V\Sigma^{-1}U^T$$