机器学习与深度学习面试系列九（降维）

降维的目的是什么?

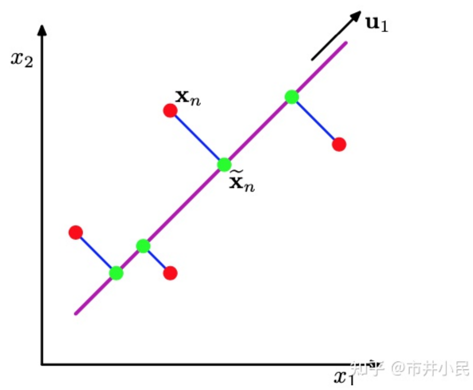
- 减少特征属性的个数，高维数据的计算复杂，同时存在维度灾难
- 方便可视化数据，高维数据的可视化很难

常见的降维方法有哪些?

主成分分析(PCA)、线性判别分析(LDA)、等距映射、局部线性嵌入、拉普拉斯特征映射、局部保留投影等。本文主要总结PCA和LDA。

什么是主成分分析?

主成分分析被定义为数据在低维线性空间上的正交投影，这个低微线性空间被称为主子空间，PCA就是找到这个主子空间。



通常可以从投影最大方差和最小重构误差两个角度来解释。

PCA和投影最大方差关系?

这个角度的直观理解是具有越大方差的方向所含的信息量越大(样本点区分的越开)。

我们先考虑D维数据投影到1维空间的情况，数据的均值为 $\hat{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$ ，投影到的1维空间为 $\{u_1\}$ ，我们只关心 u_1 的方向，并不关心其长度，不是一般性，可以定义 $u_1^T u_1 = 1$ ，即 u_1 是单位向量（坐标轴取单位向量也是符合直觉的）。

对于任何一个样本点 x_n ，投影到 u_1 上后为 $u_1^T x_n$ ，投影后的n个数据均值为 $u_1^T \hat{x}$ ，则投影后在 u_1 上的数据方差为：

$$
\begin{aligned}
J &= \frac{1}{N}\sum_{n=1}^{N}\{u_1^T x_n - u_1^T \hat{x}\}^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\{u_1^T (x_n - \hat{x})\}^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}\{u_1^T (x_n - \hat{x})(x_n - \hat{x})^T u_1\} \\
&= u_1^T \frac{1}{N}\sum_{n=1}^{N}\{(x_n - \hat{x})(x_n - \hat{x})^T\} u_1
\end{aligned}
$$

$\frac{1}{N}\sum_{n=1}^{N}\{(x_n - \hat{x})(x_n - \hat{x})^T\}$ 其实就是原始数据的协方差，记作 C ，PCA的目标就是最大化J，同时满足 $u_1^T u_1 = 1$ 。形式化为：

$$
\begin{aligned}
&\max \quad u_1^T C u_1 \\
&s.t. \quad u_1^T u_1 = 1
\end{aligned}
$$

利用拉格朗日乘子法，上面这个约束优化等价于最大化 $L = u_1^T C u_1 + \lambda(1 - u_1^T u_1)$

对 u_1 求偏导并令其等于0，得： $\frac{\partial L}{\partial u_1} = Cu_1 - \lambda u_1 = 0$ ，即 $Cu_1 = \lambda u_1$ 。

可以看出， u_1 实际上就是协方差矩阵C的特征向量， λ 为其对应的特征值。两边同时乘以 u_1^T ， $u_1^T C u_1 = J = u_1^T \lambda u_1 = \lambda$ 。也就是说，当 u_1 为协方差矩阵C的特征向量时，投影投影方差为对应的特征值。所以我们取最大的特征值对应的特征向量，叫做第一主成分。以此类推，如果我们要投影到M维空间上，我们可以前M大的特征值对应的特征向量，构成主子空间。

PCA和最小重建误差关系?

这个角度的直观理解是利用主子空间表达的样本点损失较小。

对于D维数据，如果直接考虑使用D维的单位正交向量集作为向量空间， $\{u_1, u_2 \ldots u_D\}$ ，则每个样本都可以准确的表达为： $x_n = \sum_{i=1}^{D} \alpha_{ni} u_i = \sum_{i=1}^{D} (u_i^T x_n) u_i$ 。如果我们用低维的M维单位正交向量集作为向量空间，则： $x_n \approx \sum_{m=1}^{M} \alpha_{nm} u_m = \sum_{m=1}^{M} (u_m^T x_n) u_m = \hat{x_n}$ 。

最小重建误差就是使 x_n 和 $\hat{x_n}$ 之间的距离平方最小，形式化为：

$$J=\frac{1}{N}\sum_{n}^{N}|x_n-\hat{x_n}|^2$$

$$J=\frac{1}{N}\sum_{n}^{N}|\sum_{i=1}^{D}\alpha_{ni}u_i-\sum_{m=1}^{M}\alpha_{nm}u_m|^2$$

$$J=\frac{1}{N}\sum_{n}^{N}|\sum_{i=1}^{M}\alpha_{ni}u_i+\sum_{i=M+1}^{D}\alpha_{ni}u_i-\sum_{m=1}^{M}\alpha_{nm}u_m|^2$$

$$J=\frac{1}{N}\sum_{n}^{N}|\sum_{i=M+1}^{D}\alpha_{ni}u_i|^2$$

$$J=\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}\alpha_{ni}u_i^T\sum_{j=M+1}^{D}\alpha_{nj}u_j$$

$$J=\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}\alpha_{ni}u_i^T(\alpha_{nM+1}u_{M+1}+\ldots+\alpha_{nD}u_D)$$

$$J=\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}\alpha_{ni}\alpha_{nM+1}u_i^Tu_{M+1}+\ldots+\alpha_{ni}\alpha_{nD}u_i^Tu_D$$

由于任意 u_i 和 u_j 都是正交的，且为单位向量，所有对于 $\forall i=j$ 都有 $u_i^Tu_j=1$ ，对于 $\forall i\neq j$ 都有 $u_i^Tu_j=0$ 。所以：

$$J=\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}\alpha_{ni}^2$$ ，而 $\alpha_{ni}=u_i^Tx_n$ ，所以:

$$J=\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}u_i^Tx_nx_n^Tu_i$$

$$J=u_i^T(\frac{1}{N}\sum_{n}^{N}\sum_{i=M+1}^{D}x_nx_n^T)u_i$$

$$J=u_i^TCu_i$$

此时目标形式和最大投影方差相同，区别在于此时要求最小投影方差，所以我们要去掉D-M个最小特征值对应的特征向量，这样损失就是对应的D-M个特征值之和。换句话说，就是留前M大的特征值对应的特征向量，结论与最小投影方差完全相同。

线性判别分析(Fisher判别分析)算法是怎样的?

LDA首先是为了分类服务的，因此只要找到一个投影方向 w ，使得投影后的样本尽可能按照原始类别分开，同时经常被用来对数据进行降维。相比于PCA，LDA可以作为一种有监督的降维算法。

在PCA中，算法没有考虑数据的标签(类别)，只是把原数据映射到一些方差比较大的方向上而已。LDA的思想是：**最大化类间均值，最小化类内方差**。意思就是将数据投影在低维度上，并且投影后同种类别数据的投影点尽可能的接近，不同类别数据的投影点的中心点尽可能的远。

Fisher判别分析基本思想

考虑最基本的二分类场景，对于一个有两种标签 C_1, C_2 的样本集 $\{x_n\}_{n=1}^{N}$ ，寻找一个方向 w ，使得样本集在这上面的投影满足类间均值差最大，类内方差最小。

对于两类样本的均值分别是 $m_1 = \frac{1}{N_1}\sum\limits_{x_n \in C_1} x_n$ ， $m_2 = \frac{1}{N_2}\sum\limits_{x_n \in C_2} x_n$ ，投影到 w 方向后，在 w 上的两类投影点的均值分别为 $\hat{m}_1 = w^T m_1$ ， $\hat{m}_2 = w^T m_2$ ，类间均值差最大就是最大化 $\hat{m}_1 - \hat{m}_2$ 。投影后的方差为： $s_1^2 = \frac{1}{N_1}\sum\limits_{x_n \in C_1}(w^T x_n - \hat{m}_1)^2$ ，$s_2^2 = \frac{1}{N_2}\sum\limits_{x_n \in C_2}(w^T x_n - \hat{m}_2)^2$ ，类内方差最小就是最小化 $s_1^2 + s_2^2$ 。要同时满足这两个条件，我们定义一个函数： $J = \frac{(\hat{m}_1 - \hat{m}_2)^2}{s_1^2 + s_2^2}$ ，使 J 最大化，就可以同时满足上述两个条件。

$$J = \frac{(\hat{m}_1 - \hat{m}_2)^2}{s_1^2 + s_2^2}$$
$$J = \frac{w^T(m_1 - m_2)(m_1 - m_2)^T w}{w^T \frac{1}{N_1}\sum_{x_n \in C_1}(x_n - m_1)(x_n - m_1)^2 w + w^T \frac{1}{N_2}\sum_{x_n \in C_2}(x_n - m_2)(x_n - m_2)^2 w}$$
$$J = \frac{w^T(m_1 - m_2)(m_1 - m_2)^T w}{w^T(\frac{1}{N_1}\sum_{x_n \in C_1}(x_n - m_1)(x_n - m_1)^T + \frac{1}{N_2}\sum_{x_n \in C_2}(x_n - m_2)(x_n - m_2)^T)w}$$

为了表达简便，记： $S_B = (m_1 - m_2)(m_1 - m_2)^T$ ，

$S_W = \frac{1}{N_1}\sum\limits_{x_n \in C_1}(x_n - m_1)(x_n - m_1)^T + \frac{1}{N_2}\sum\limits_{x_n \in C_2}(x_n - m_2)(x_n - m_2)^T$ ，

则： $J = \frac{w^T S_B w}{w^T S_W w}$ ，要使 J 最小，我们求 w 的偏导，并令其等于0，得：

$\frac{\partial J}{\partial w} = \frac{(w^T S_W w) S_B w - (w^T S_B w) S_W w}{(w^T S_W w)^2} = 0$ ，即分子等于0:

$$
\begin{aligned}
&(w^T S_W w) S_B w = (w^T S_B w) S_W w \\
&S_B w = \frac{(w^T S_B w)}{(w^T S_W w)} S_W w \\
&S_B w = \lambda S_W w \\
&S_W^{-1} S_B w = \lambda w
\end{aligned}
$$

其中我们令 $\lambda = \dfrac{w^T S_B w}{w^T S_W w} = J$， λ 的值和 J 相等，而从上式可以看出， w 是 $S_W^{-1} S_B$ 矩阵的特征向量， λ 是其对应的特征值。要使 J 最大，我们只需求 $S_W^{-1} S_B$ 矩阵最大特征值对应的特征向量即可。

LDA如何支持多分类?

假设要求k分类，则

$$S_B = \sum_{j=1}^{k} N_j (m_j - m)(m_j - m)^T$$ ，其中 m 为所有样本点的均值。

$$S_W = \sum_{j=1}^{k} \frac{1}{N_j} \sum_{x_n \in C_j} (x_n - m_j)(x_n - m_j)^T$$

求解方式和上述完全相同，只需求 $S_W^{-1} S_B$ 矩阵最大若干个特征值对应的特征向量即可。

由于 S_B 矩阵的秩为k-1(第k行可以由前k-1行线性组合得到)，所以 $S_W^{-1} S_B$ 矩阵最多有k-1个特征值，也就是说LDA方法最多降到类别数k-1的维数。

LDA算法的优缺点?

优点

1. 在降维过程中可以使用类别的先验知识经验，而像PCA这样的无监督学习则无法使用类别先验知识。
2. LDA在样本分类信息依赖均值而不是方差的时候，比PCA之类的算法较优。

缺点

1. LDA不适合对非高斯分布样本进行降维，PCA也有这个问题。
2. LDA降维最多降到类别数k-1的维数，如果我们降维的维度大于k-1，则不能使用LDA。当然目前有一些LDA的进化版算法可以绕过这个问题。

3. LDA在样本分类信息依赖方差而不是均值的时候，降维效果不好。
4. LDA可能过度拟合数据。

PCA和LDA区别和联系？

相同点

1. 两者均可以对数据进行降维。
2. 两者在降维时均使用了矩阵特征分解的思想。
3. 两者都假设数据符合高斯分布。

不同点

1. LDA是有监督的降维方法，而PCA是无监督的降维方法
2. LDA降维最多降到类别数k-1的维数，而PCA没有这个限制。
3. LDA除了可以用于降维，还可以用于分类。
4. LDA选择分类性能最好的投影方向，而PCA选择样本点投影具有最大方差的方向。举一个简单的例子，在语音识别中，我们想从一段音频中提取出人的语音信号，这时可以使用PCA先进行降维，过滤掉一些固定频率(方差较小)的背景噪声。但如果我们的需求是从这段音频中区分出声音属于哪个人，那么我们应该使用LDA对数据进行降维，使每个人的语音信号具有区分性。