降维

在很多机器学习问题中，训练集中的每条数据经常伴随着上千、甚至上万个特征。要处理这所有的特征的话，不仅会让训练非常缓慢，还会极大增加搜寻良好解决方案的困难。这个问题就是我们常说的维度灾难。不过值得庆幸的是，在实际问题中，经常可以极大地减少特征的数目，将棘手的问题转变为容易处理的问题。

维度灾难

在高维空间中，很多事情的行为都会非常不一样。例如，假设我们在一个单元正方形（1x1正方形）中选择一个随机点，则此点仅有40%的概率与边框的距离小于0.001（也就是说，一个随机点不太可能非常靠近某个维度）。但是在一个10000维的单元超立方体中，这个概率要高于99.999999%。**大部分在高维超平面中的点都非常接近于边界。**

还有一个更麻烦的差异：假设我们在一个单元正方形中随机选取两个点，这两个点的平均距离约为0.52。如果我们在一个3D立方体中随机选择两个点，则平均距离大约为0.66。但是如果我们在1000000维超立方体中随机选择两个点的话，它们的平均距离大约为408.25（约为1000000/6的平方根 ）。这是一个很反直觉的现象：为什么两个点都在同样的单元超平面中，但是距离可以离的这么远？当然这是由于在高维中有足够多的空间导致了。所以这样导致的结果就是：**高维数据集中的数据点可能会非常稀疏（或离散）**。大多数训练实例可能相互之间离的都非常远，导致预测性能相对于低维数据集来说会更不可靠，因为它们基于的是更大的外推法（extrapolations）。简单地说，训练集的维度越高，过拟合的风险越大。

降维算法

降维算法主要依赖于以下两个降维方法：投影（projecting）与流形学习（Manifold Learning）。

主成成分分析（PCA）

主成成分分析（Principal Component Analysis，PCA）在目前是非常热门的降维算法。首先它找到一个最接近数据的超平面，然后将数据投影到这个平面上。假设有数据集$\{x^1, x^2, \dots, x^m\}, x^i \in R^n$，我的目的是将数据维度减低到$k, k < n$。

数据预处理：

- Set $\mu = \frac{1}{m}\sum_{i=1}^{m} x^i$
- Replace x^i with $x^i - \mu$
- Set $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^i)^2$
- Replace x_j^i with $\frac{x_j^i}{\sigma_j}$

数据x在低维空间的投影表示为：$x^{iT}u$，选择$||u|| = 1$，为了使数据在低维空间尽量保留原来的分布规律，我们的目标函数设为

$$\max_{||u||=1} \frac{1}{m}\sum_{i=1}^{m} |x^{iT}u|^2 = u^T \left[\frac{1}{m}\sum_{i=1}^{m} x^i x^{iT}\right] u = u^T \Sigma u$$

Σ是x的谐方差矩阵，利用拉格朗日乘子法，可以得到u的每一列都是Σ的本征态。

降维：选择前k个特征向量$u_1, u_2, \dots, u_k$，降维后的数据可以表示为，

$$y^i = \{u_1^T x^i, u_2^T x^i, \dots, u_k^T x^i\}$$

当维度n很大时，直接对协方差矩阵求本征态是不方便的。所以，我们一般可以对x做奇异值分解，得到本征向量。

奇异值分解（SVD）

假设有一个矩阵$A \in R^{m \times n}$，可以分解为

$$A = UDV^T, U \in R^{m \times n}, D = Diag(\sigma_1, \sigma_2, \ldots, \sigma_n) \in R^{n \times n}, V \in R^{n \times n}$$

其中：

- U的列：AA^T的本征态
- V的列：$A^T A$的本征态

即$x = UDV^T$，选择V中的前k列作为协方差矩阵的特征向量。

总结

无监督算法的应用方法

数据分布	建模概率P(x)	无法得到概率
子空间	因子分析模型	PCA模型
成块或聚团	混合高斯模型	K-means模型