0、概述

样本数据（的特征的）降维，一般作为特征工程的一部分，用于样本数据输入模型之前处理；同时它本身也可算作一种机器学习模型——一种无监督的机器学习模型，因为它需要训练数据，它需要了解数据的结构（或者说分布的信息），才能针对性地有效地降维。

降维的本质是将数据从高维特征空间映射到低维特征空间，并使数据在低维空间尽量保持其在高维空间中的某种意义的“结构”或者说分布特征。

为什么要将数据映射到低维特征空间呢？因为在实际中，采集的数据虽然在高维空间，但常常因为数据的信息冗余（特征之间的相关性所造成的）和噪声的原因，大部分集中在某个低维的子空间。出于减少冗余和降噪的目的，我们就应当将数据映射到低维空间。

降维还可以有助于实现数据的可视化——当数据被降维到二维或三维时，就可以在图中显示出来，直观地进行观察和分析。

实际中的数据各式各样，它们在空间中的分布特点都不一样，因此，针对不同的数据分布特点，有很多种降维算法被提出。降维算法可以分为线性降维和非线性降维。

实际中，降维算法的实现一般都将样本数据看成是一个矩阵（），然后针对该矩阵进行计算。矩阵的每一行是一个样本，每一列是一个特征。

常用的降维算法有：PCA、KPCA、FA、流形学习等。下面一一介绍它们。

**1、PCA**

PCA是最经典的线性降维算法。那么，“线性降维”中的“线性”是怎么体现的呢？就是说，PCA用一个“映射矩阵”去乘样本数据，就得到了降维的结果，这相当于对样本数据执行了一个线性的坐标变换（“坐标”其实就是特征）。

我们知道，机器学习模型一般都是基于一个假设，导出其目标函数，然后对目标函数进行最优化求解。PCA也不例外。PCA的假设是，样本数据的各维特征之间具有线性相关性（这造成了数据的信息冗余），并且数据中存在一些高斯噪声，显然，通过线性映射将数据映射到低维空间，可以减小特征之间的线性相关性。那么自然地，PCA的目标就是：将数据线性映射到低维空间，尽量消除特征之间的信息冗余（也就是去线性相关性），同时要尽量保持数据信息不被丢失。（在实现去线性相关的过程中，恰好能顺便压制噪声）。

注意，这里所说的线性相关性是指特征之间的线性相关性。这里，特征被看做是随机变量，随机变量的“线性相关性”是概率论中的内容，它与线性代数中的“线性相关性”是有区别的，它们的研究对象不同。

这个目标有很多的形式化的方法，比如：最大方差目标、最小均方误差目标等，这里我们只介绍最大方差目标。

最大方差法的基本思想是用方差衡量数据的信息量，以便对给定的降维，能够知道其保留了原始数据的多少信息。

设样本数据为（m个样本，n维特征），最大方差法首先将数据线性映射到某一维特征空间（或者说一维坐标轴）上，并保证在该维上方差最大。为什么要使方差最大？因为最大的方差对应着最多的信息。对于一个随机变量，我们最常用熵来衡量它的信息量（或者说不确定程度），实际上，方差也可以衡量随机变量的信息量，方差越大，信息量越大。

将n维特征的数据映射到一维，必然会损失信息，而最大方差的映射可以保证信息损失最小。

然后，选择与该一维坐标轴垂直的另一维坐标轴，使（除了以外）原始数据在上的投影的方差最大。为什么要与垂直呢？因为这样就能保证，假如需要将原始数据线性降维成二维，则和就是能保证方差最大的两个维度的坐标轴。

然后再重复地选取与前面选定的坐标轴都垂直的坐标轴，并使原始数据在该维坐标轴上映射的方差最大化。

最终可以选出来n个坐标轴（n是原始数据的特征维数）。这些坐标轴互相垂直，它们按照原始数据在其上映射的方差大小从高到低排序。因此，选择前k（k需要人为指定）个坐标轴，将原始数据映射到这k个坐标轴构成的坐标系中，就能保证原始数据的信息得到尽量大的保留（k越大，保留的信息越多）。这样就实现了方差最大化的的降维。

另外，在我们舍弃的那些坐标轴方向上，常常是数据的噪声（因为它们的方差很小），所以PCA还能在一定程度上起到去除噪声的作用。

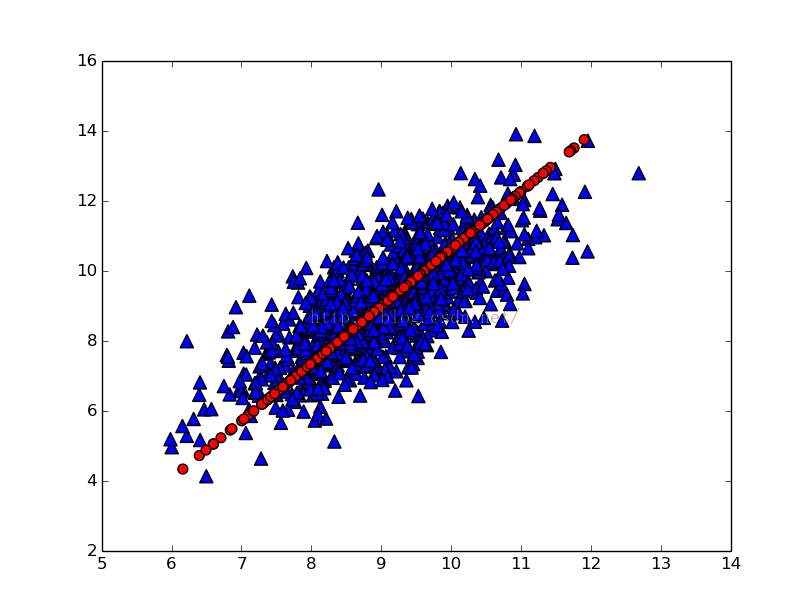


图1 PCA示意图

可以证明，上述求取PCA的映射坐标的过程，等价于求解（零均值化后的）原始数据矩阵的协方差矩阵的特征向量的过程。每一个特征向量就是一维映射坐标轴。特征向量对应的特征值的大小就是原始数据在该维坐标轴上映射的方差的大小。

由此得到PCA算法的具体流程：

（1）对原始数据矩阵求协方差矩阵；

（2）对协方差矩阵C进行特征值分解，将得到的特征值按从大到小顺序排序：，对应的特征向量也作相应排序：；

（3）选取前k个特征向量，构成映射矩阵，是n×k的矩阵；

（4）用去乘原始数据矩阵X，就得到映射后的数据；对某一个样本，也是用去乘它，得到样本映射到低维空间中的结果。

**2、KPCA**

KPCA（核主成分分析）是在PCA的基础上引入核技巧，而将PCA扩展为适用于非线性降维的算法。

PCA是对原始数据作线性的映射（坐标变换），KPCA则是对原始数据作非线性的映射。

我们知道，当一份数据可以看作是低维空间中的数据嵌入到高维空间中时，就可以对它进行降维。PCA能够对线性嵌入到高维空间中的数据有效地降维；但对于非线性嵌入到高维空间中的数据，它就无能为力了。

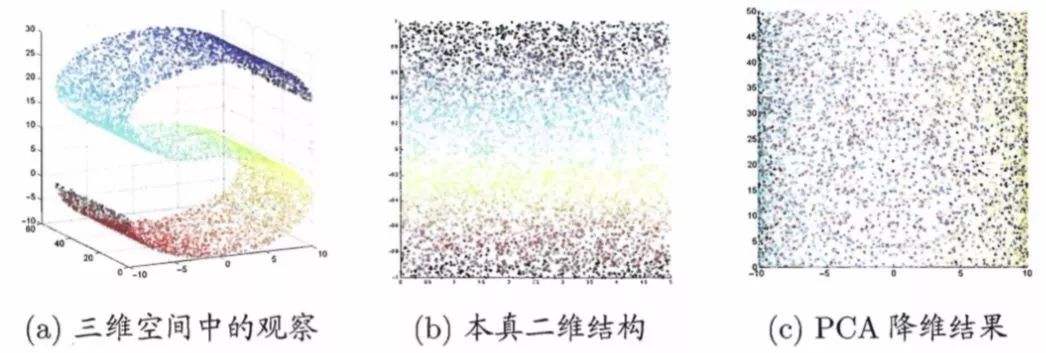


图2 非线性嵌入高维空间的数据的降维

如图2所示，一个二维的数据以图2(a)的方式嵌入三维空间，其本来的数据分布（二维）如图2(b)，但如果使用PCA将其映射到二维空间，结果如图2(c)，显然不是我们想要的结果。

对于这种问题，我们可能使用PCA进行降维，选取前几个较大的特征值，使这些特征值的贡献度达到95%以上，然后做映射，我们就觉得其结果已经保留了原始数据的绝大部分信息，good enough。但这是不对的，因为出发点就错了：PCA作为线性映射，试图保留原始数据的点与点之间的高维空间中的欧氏距离大小的相对关系，但是我们真正想要的是保留其“内蕴距离”的大小。因此PCA可以说确实保留了大部分原始数据的信息，但是信息本身是错的，这就如缘木求鱼，南辕北辙，最终只能竹篮打水一场空。

KPCA解决这种问题的思路是，先将数据映射到更高维空间（此时数据就变成了线性嵌入），然后再执行PCA。

但在实际问题中，原始数据特征空间到更高维空间的映射函数很难给出，因此我们引入核函数，采取一种间接的方法计算KPCA。

下面详细介绍KPCA的推导过程。

假设有一个映射，它将样本映射到高维空间：



其中是样本集中第i个样本，是该样本映射到高维空间中的向量。

则理想情况下，KPCA的流程就是：先根据映射将原始数据映射到高维空间，然后再执行PCA，执行PCA主要就是对协方差矩阵特征值分解：

 （1）

这里将高维空间中的数据的协方差矩阵写成基于（高维空间中）样本的表示式，方便后面推导。表示高维空间数据协方差矩阵特征向量，这些特征向量构成d维正交空间。

（1）式也可以写成：

 （2）

可以看出，若有的具体形式，则很容易执行KPCA。但是实际问题中我们通常很难知道的形式。为此我们引入核函数：

 （3）

可以看出，核函数是关于任意两个样本的。核函数其实就是两个样本映射到高维空间中的向量的内积。

又根据（1）式可以得到：

 （4）

将（4）式和（3）式带入（2）式，化简，并用矩阵和向量来表示，有：

 （5）

其中，（向量），其每个分量（标量）。

K是核矩阵，其元素值由两个样本的核函数值给出：。

式（5）是一个特征值分解问题，解之得。那么解出有什么用呢？原来啊，它是用来计算最终的映射结果的——对某个样本x，其（经过两次映射）最终结果的第j维坐标是：

 （6）

或者写成向量的形式：

 （7）

式中，是（5）式求解特征值分解问题解出的特征向量，也是个向量，，进一步，可以直接写成从样本x到最终映射结果向量的式子：

 （8）

其中是最终降维结果（向量），其维数是（事先选定的数）：；是映射矩阵，是从（5）式解出的特征向量中，选取前个特征向量构成的矩阵：。

为什么最终结果没有涉及到原始数据映射到的高维空间的维数d呢？在上面所述的流程中，高维空间的维数d的取值到底是多少呢？实际上，d可以取小于m的任意值（这里强调一下，d必须小于样本集中的样本个数m），d取不同的值不会影响最终结果，当然，d是要大于n（原始数据特征数）的。

实际上，可以这样理解：在上述的基于核函数的KPCA中，高维特征空间的维数就是指定为m的。

总的来说，KPCA通过引入核函数，避免了计算原始样本到高维空间的映射。注意，不能将核矩阵K当作成是样本到高维空间中的映射，K只是绕过计算样本到高维映射的一个手段。

**3、FA**

这里大概介绍一下FA的思想，不对其详细说明。

FA（因子分析）是统计学方法的一种。它的思想是，在原始特征中，特征之间可能存在着相关性。这种相关性的表现是，特征可以分为几部分，每部分中各特征之间具有较强的相关性，而各部分之间相关性较弱。基于这种理解，FA的目标是对每部分特征重构出一个或几个具有代表性的特征，作为该部分特征的“公共因子”。

举一个实际问题的例子，我们测量多个学生的各科成绩：数学、语文、英语、100米，铅球，跳高。发现学生的数学、语文、英语成绩具有很强的相关性，100米、铅球、跳高也具有很强的相关性，但是数学和100米，语文和铅球等，相关性比较差。这时可以提取因子“智力成绩”来代表数学、语文、英语成绩；提取“体力因子”代表100米、铅球、跳高成绩，这就是所谓的公共因子。

具体的FA方法有很多种。

FA与PCA、KPCA的区别是：后者的重点在于解释各变量的总方差，而前者则把重点放在解释各变量之间的协方差；后者降维后的结果可解释性不强，前者的结果具有更强的可解释性。

**4、流形学习**

流形学习将样本集看成是在嵌入到高维空间中的一个低维流形上进行采样而得的。

什么是流形？流形是微分几何中的概念，是指（整体上）不一定是欧氏空间，但局部具有欧氏空间性质的几何体。

嵌入到高维空间中的低维流形，在流形上的任意两个点在流形上的距离（内蕴距离），与这两个点在高维空间中的欧氏距离，一般来说是不相等的，但是在局部上，可以用欧氏距离来代替内蕴距离。

（内蕴距离，也叫测地线距离，是指两点在流形上的最短路径的长度，对于欧氏空间来说，内蕴距离就是欧氏距离）

举个例子：二维球面就是一个嵌入到三维空间中的二维流形，它本身不是平坦的，但是局部上看是平坦的，局部上的距离可以用其切面的距离（也就是欧氏距离）代替。

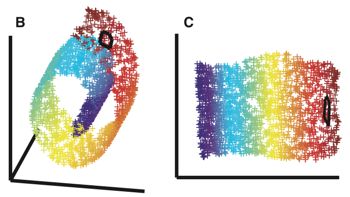


图3 低维流形嵌入到高维空间示意图

如图3所示，左图为一个二维流形嵌入到三维空间，右图是该二维流形 “展平”回二维的结构。这种“展平”的投影方式，实际上相当于用该流形的内蕴坐标来表示它。这也就起到降维的作用。

基于对数据集的这种理解，流形学习的降维方法的目标就是找到数据集对应的的低维流形结构，并将其投影回低维空间，以达到降维之目的。

在流形学习中，我们常常要计算在数据集中两个样本的测地距离，如何计算呢？一般是通过图的方法来计算，对样本集构建“近邻连接图”，在图上计算两点之间的最短路径（使用Dijkstra算法）。

流形学习有许多具体的算法，它们的具体思路是将高维数据投影到低维空间，使低维的数据能保持对应的高维数据中的流形结构。

下面简单地介绍几种常用的流形学习降维算法的基本思路：

（1）Isomap（等度量映射）：在低维空间中，尽量保持所有点之间（在高维空间中）的测地距离的大小关系；

（2）LLE（局部线性嵌入）：在低维空间中，尽量保持（在高维空间中）邻域内样本之间的线性关系；

（3）LE（拉普拉斯特征映射）：Isomap是保持所有点的测地距离，是一种全局算法，而拉普拉斯特征映射是保持局部邻接关系，即在高维空间中距离较近的样本，在低维空间中也尽量保持靠近，它是一种局部算法。

除以上内容之外，还有一些重要内容需要补充：奇异值分解与PCA的关系、SOM降维算法、谱聚类与拉普拉斯特征映射的关系等。