PCA原理与实践

# 什么是PCA

PCA（principal component analysis，主成分分析），是一种数据降维的方法，利用正交变换 (orthogonal transformation)把一系列可能线性相关的变量转换为一组线性不相关的新变量，也称为主成分，从而将高维数据转换为低维数据，降低模型训练所需要的计算资源，通常用于数据预处理阶段。PCA是一种降维的经典算法，属于线性、无监督、全局的降维方法。

主成分是原有变量的线性组合，其数目不多于原始变量。组合之后，相当于我们获得了一批新的观测数据，这些数据的含义不同于原有数据，但包含了之前数据的大部分特征，并且有着较低的维度，便于进一步的分析。

在空间上，PCA可以理解为把原始数据投射到一个新的坐标系统，第一主成分为第一坐标轴，它的含义代表了原始数据中多个变量经过某种变换得到的新变量的变化区间；第二成分为第二坐标轴，代表了原始数据中多个变量经过某种变换得到的第二个新变量的变化区间。这样我们把利用原始数据解释样品的差异转变为利用新变量解释样品的差异。

这种投射方式会有很多，为了最大限度保留对原始数据的解释，一般会用最大方差理论或最小损失理论，使得第一主成分有着最大的方差或变异数 (就是说其能尽量多的解释原始数据的差异)；随后的每一个主成分都与前面的主成分正交，且有着仅次于前一主成分的最大方差 (正交简单的理解就是两个主成分空间夹角为90°，两者之间无线性关联，从而完成去冗余操作)。

|  |
| --- |
| 降维有两个主要算法：线性判别分析（Linear Discriminant Analysis，简称LDA）和主成分分析。这两者之间最根本的区别在于，LDA利用类的信息去找新特征，以最大化类的可分离性，而PCA利用每个特征的方差最大化不同特征的可分离性。因此，LDA是有监督算法，而PCA是无监督算法。 |

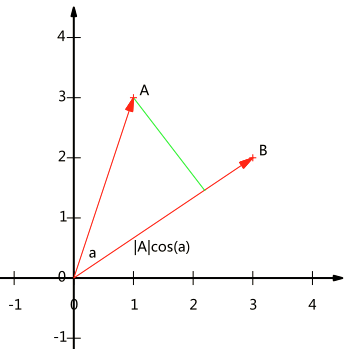
# 前置知识

## 内积与投影

两个维数相同的向量的内积被定义为：



内积运算将两个向量映射为一个实数。其计算方式非常容易理解，但是其意义并不明显。下面我们分析内积的几何意义。假设A和B是两个n维向量，我们知道n维向量可以等价表示为n维空间中的一条从原点发射的有向线段，为了简单起见我们假设A和B均为二维向量，A=(x1,y1)，B=(x2,y2)。则在二维平面上A和B可以用两条发自原点的有向线段表示，如下图：



我们从A点向B所在直线引一条垂线。垂线与B的交点叫做A在B上的投影，再设A与B的夹角是a，则投影的矢量长度为|A|cos(a)，其中是向量A的模，也就是A线段的标量长度。

这里我们需要区分矢量长度和标量长度，标量长度总是大于等于0，值就是线段的长度；而矢量长度可能为负，其绝对值是线段长度，而符号取决于其方向与标准方向相同或相反。

到这里还是看不出内积和这东西有什么关系，不过如果我们将内积表示为另一种我们熟悉的形式：



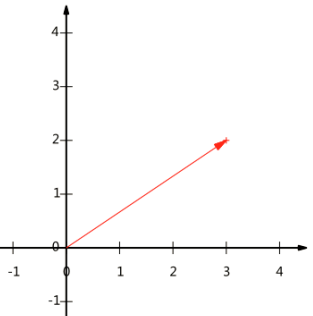
A与B的内积等于A到B的投影长度乘以B的模。再进一步，如果假设B的模为1，即让|B|=1，那么就变成了：



也就是说，设向量B的模为1，则A与B的内积值等于向量A在向量B所在直线投影的矢量长度！这就是内积的一种几何解释，也是我们得到的第一个重要结论。在后面的推导中，将反复使用这个结论。

## 向量的基

一个二维向量可以对应二维笛卡尔直角坐标系中从原点出发的一个有向线段。例如在二维空间中的向量：



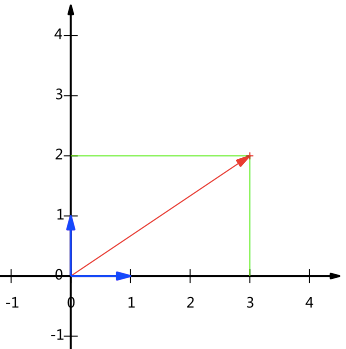
我们经常用线段终点的点坐标表示向量，例如上面的向量可以表示为(3,2)。

然而，只有一个(3,2)本身是不能够精确表示一个向量的。我们仔细看一下，这里的3实际表示的是向量在x轴上的投影值是3，在y轴上的投影值是2。也就是说我们其实隐式引入了一个定义：以x轴和y轴上正方向长度为1的向量为标准。那么一个向量(3,2)实际是说在x轴投影为3，而y轴的投影为2。注意投影是一个矢量，所以可以为负。

更正式的说，向量(x,y)实际上表示线性组合：



不难证明所有二维向量都可以表示为这样的线性组合。此处(1,0)和(0,1)叫做二维空间中的一组基。



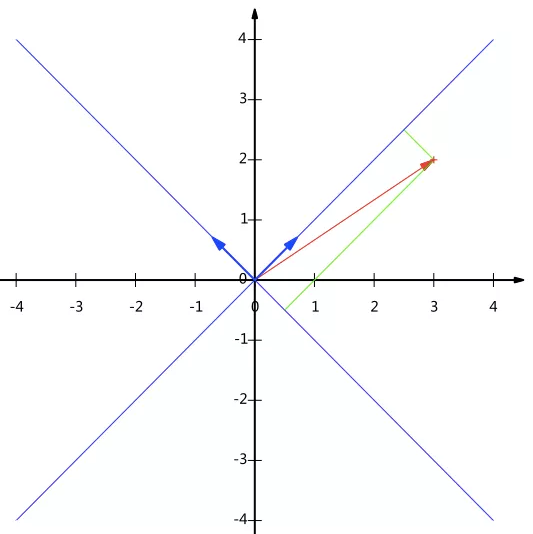
所以，要准确描述向量，首先要确定一组基，然后给出在基所在的各个直线上的投影值，就可以了。只不过我们经常省略第一步，而默认以(1,0)和(0,1)为基。

之所以默认选择(1,0)和(0,1)为基，当然是比较方便，因为它们分别是x和y轴正方向上的单位向量，因此就使得二维平面上点坐标和向量一一对应，非常方便。但实际上任何两个线性无关的二维向量都可以成为一组基，所谓线性无关在二维平面内可以直观认为是两个不在一条直线上的向量。

例如，(1,1)和(-1,1)也可以成为一组基。一般来说，我们希望基的模是1，因为从内积的意义可以看到，如果基的模是1，那么就可以方便的用向量点乘基而直接获得其在新基上的坐标了！实际上，对应任何一个向量我们总可以找到其同方向上模为1的向量，只要让两个分量分别除以模就好了。例如，上面的基可以变为。

现在，我们想获得(3,2)在新基上的坐标，即在两个方向上的投影矢量值，那么根据内积的几何意义，我们只要分别计算(3,2)和两个基的内积，不难得到新的坐标为图片

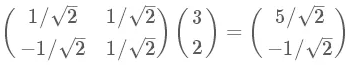
下图给出了新的基以及(3,2)在新基上坐标值的示意图：



另外这里要注意的是，上面列举的例子中基是正交的（即内积为0，或直观说相互垂直），但可以成为一组基的唯一要求就是线性无关，非正交的基也是可以的。不过因为正交基有较好的性质，所以一般使用的基都是正交的。

## 基变换的矩阵表示

下面我们找一种简便的方式来表示基变换。还是拿上面的例子，想一下，将(3,2)变换为新基上的坐标，就是用(3,2)与第一个基做内积运算，作为第一个新的坐标分量，然后用(3,2)与第二个基做内积运算，作为第二个新坐标的分量。实际上，我们可以用矩阵相乘的形式简洁的表示这个变换：



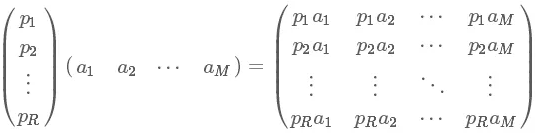
太漂亮了！其中矩阵的两行分别为两个基，乘以原向量，其结果刚好为新基的坐标。可以稍微推广一下，如果我们有m个二维向量，只要将二维向量按列排成一个两行m列矩阵，然后用“基矩阵”乘以这个矩阵，就得到了所有这些向量在新基下的值。例如(1,1)，(2,2)，(3,3)，想变换到刚才那组基上，则可以这样表示：



于是一组向量的基变换被干净的表示为矩阵的相乘。

一般的，如果我们有M个N维向量，想将其变换为由R个N维向量表示的新空间中，那么首先将R个基按行组成矩阵A，然后将向量按列组成矩阵B，那么两矩阵的乘积AB就是变换结果，其中AB的第m列为A中第m列变换后的结果。

数学表示为：



其中pi是一个行向量，表示第i个基，aj是一个列向量，表示第j个原始数据记录。

特别要注意的是，这里R可以小于N，而R决定了变换后数据的维数。也就是说，可以将一N维数据变换到更低维度的空间中去，变换后的维度取决于基的数量。因此这种矩阵相乘的表示也可以表示降维变换。

最后，上述分析同时给矩阵相乘找到了一种物理解释：两个矩阵相乘的意义是将右边矩阵中的每一列列向量变换到左边矩阵中每一行行向量为基所表示的空间中去。更抽象的说，一个矩阵可以表示一种线性变换。很多同学在学线性代数时对矩阵相乘的方法感到奇怪，但是如果明白了矩阵相乘的物理意义，其合理性就一目了然了。

# 协方差矩阵及优化目标

上面讨论了选择不同的基可以对同样一组数据给出不同的表示，而且如果基的数量少于向量本身的维数，则可以达到降维的效果。但是我们还没有回答一个最最关键的问题：如何选择基才是最优的。或者说，如果我们有一组N维向量，现在要将其降到K维（K小于N），那么我们应该如何选择K个基才能最大程度保留原有的信息？

要完全数学化这个问题非常繁杂，这里用一种非形式化的直观方法来看这个问题。

为了避免过于抽象的讨论，我们仍以一个具体的例子展开。假设我们的数据由五条记录组成，将它们表示成矩阵形式：

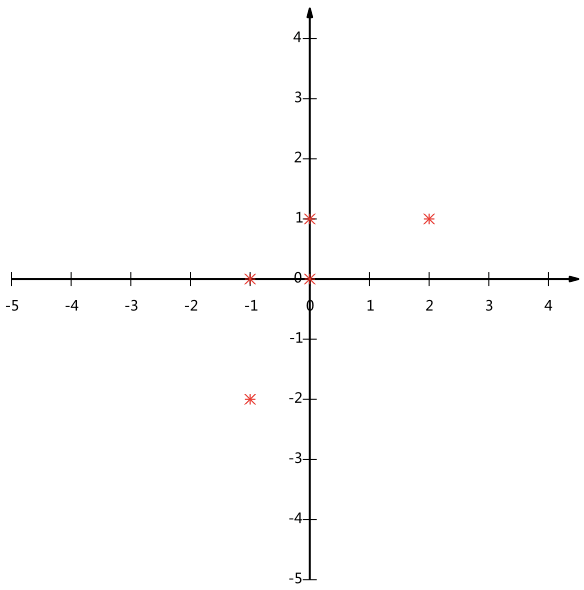


其中每一列为一条数据记录，而一行为一个字段。为了后续处理方便，我们首先将每个字段内所有值都减去字段均值，其结果是将每个字段都变为均值为0（这样做的道理和好处后面会看到）。

我们看上面的数据，第一个字段均值为2，第二个字段均值为3，所以变换后：



五条数据在平面直角坐标系内的样子如下图：



现在问题来了：如果我们必须使用一维来表示这些数据，又希望尽量保留原始的信息，你要如何选择？

通过上一节对基变换的讨论我们知道，这个问题实际上是要在二维平面中选择一个方向，将所有数据都投影到这个方向所在直线上，用投影值表示原始记录。这是一个实际的二维降到一维的问题。

那么如何选择这个方向（或者说基）才能尽量保留最多的原始信息呢？一种直观的看法是：希望投影后的投影值尽可能分散。

以上图为例，可以看出如果向x轴投影，那么最左边的两个点会重叠在一起，中间的两个点也会重叠在一起，于是本身四个各不相同的二维点投影后只剩下两个不同的值了，这是一种严重的信息丢失，同理，如果向y轴投影最上面的两个点和分布在x轴上的两个点也会重叠。所以看来x和y轴都不是最好的投影选择。我们直观目测，如果向通过第一象限和第三象限的斜线投影，则五个点在投影后还是可以区分的。

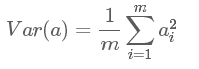
下面，我们用数学方法表述这个问题。

## 方差

上文说到，我们希望投影后投影值尽可能分散，而这种分散程度，可以用数学上的方差来表述。此处，一个字段的方差可以看做是每个元素与字段均值的差的平方和的均值，即：



由于上面我们已经将每个字段的均值都化为0了，因此方差可以直接用每个元素的平方和除以元素个数表示：



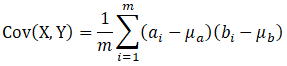
于是上面的问题被形式化表述为：寻找一个一维基，使得所有数据变换为这个基上的坐标表示后，方差值最大。

## 协方差

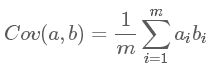
对于上面二维降成一维的问题来说，找到那个使得方差最大的方向就可以了。不过对于更高维，还有一个问题需要解决。考虑三维降到二维问题。与之前相同，首先我们希望找到一个方向使得投影后方差最大，这样就完成了第一个方向的选择，继而我们选择第二个投影方向。

如果我们还是单纯只选择方差最大的方向，很明显，这个方向与第一个方向应该是“几乎重合在一起”，显然这样的维度是没有用的，因此，应该有其他约束条件。从直观上说，让两个字段尽可能表示更多的原始信息，我们是不希望它们之间存在（线性）相关性的，因为相关性意味着两个字段不是完全独立，必然存在重复表示的信息。

数学上可以用两个字段的协方差表示其相关性，相关性的公式如下：



由于已经让每个字段均值为0，因此，协方差可以简化为：



可以看到，在字段均值为0的情况下，两个字段的协方差简洁的表示为其内积除以元素数m。

当协方差为0时，表示两个字段完全独立。为了让协方差为0，我们选择第二个基时只能在与第一个基正交的方向上选择。因此最终选择的两个方向一定是正交的。

至此，我们得到了降维问题的优化目标：将一组N维向量降为K维（K大于0，小于N），其目标是选择K个单位（模为1）正交基，使得原始数据变换到这组基上后，各字段两两间协方差为0，而字段的方差则尽可能大（在正交的约束下，取最大的K个方差）。

## 协方差矩阵

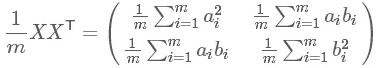
上面我们导出了优化目标，但是这个目标似乎不能直接作为操作指南（或者说算法），因为它只说要什么，但根本没有说怎么做。所以我们要继续在数学上研究计算方案。

我们看到，最终要达到的目的与字段内方差及字段间协方差有密切关系。因此我们希望能将两者统一表示，仔细观察发现，两者均可以表示为内积的形式，而内积又与矩阵相乘密切相关。于是我们来了灵感：

假设我们只有a和b两个字段，那么我们将它们按行组成矩阵X：



然后用X乘以X的转置，并乘上系数1/m：



奇迹出现了！这个矩阵对角线上的两个元素分别是两个字段的方差，而其它元素是a和b的协方差。两者被统一到了一个矩阵的。

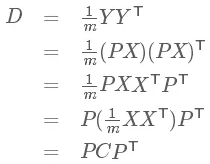
根据矩阵相乘的运算法则，这个结论很容易被推广到一般情况：

设有m个n维数据记录，将其按列排成n乘m的矩阵X，设，则C是一个对称矩阵，其对角线分别个各个字段的方差，而第i行j列和j行i列元素相同，表示i和j两个字段的协方差。

## 协方差矩阵对角化

根据上述推导，我们发现优化的结果等价于将协方差矩阵对角化：即除对角线外的其它元素化为0，并且在对角线上将元素按大小从上到下排列，这样我们就达到了优化目的。这样说可能还不是很明晰，我们进一步看下原矩阵与基变换后矩阵协方差矩阵的关系：

设原始数据矩阵X对应的协方差矩阵为C，而P是一组基按行组成的矩阵，设Y=PX，则Y为X对P做基变换后的数据。设Y的协方差矩阵为D，我们推导一下D与C的关系：



现在事情很明白了！我们要找的P不是别的，而是能让原始协方差矩阵对角化的P。换句话说，优化目标变成了寻找一个矩阵P，使是一个对角矩阵，并且对角元素按从大到小依次排列，那么P的前K行就是要寻找的基，用P的前K行组成的矩阵乘以X就使得X从N维降到了K维，并满足上述优化条件。

至此，我们离“发明”PCA还有仅一步之遥！

现在所有焦点都聚焦在了协方差矩阵对角化问题上，有时，我们真应该感谢数学家的先行，因为矩阵对角化在线性代数领域已经属于被玩烂了的东西，所以这在数学上根本不是问题。

由上文知道，协方差矩阵C是一个是对称矩阵，在线性代数上，实对称矩阵有一系列非常好的性质：

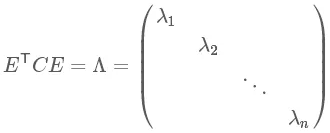
1）实对称矩阵不同特征值对应的特征向量必然正交。

2）设特征向量λ重数为r，则必然存在r个线性无关的特征向量对应于λ，因此可以将这r个特征向量单位正交化。

由上面两条可知，一个n行n列的实对称矩阵一定可以找到n个单位正交特征向量，设这n个特征向量为e1,e2,⋯,en，我们将其按列组成矩阵：



则对协方差矩阵C有如下结论：



其中Λ为对角矩阵，其对角元素为各特征向量对应的特征值（可能有重复）。

以上结论不再给出严格的数学证明，对证明感兴趣的朋友可以参考线性代数书籍关于“实对称矩阵对角化”的内容。

到这里，我们发现我们已经找到了需要的矩阵P:

图片

P是协方差矩阵的特征向量单位化后按行排列出的矩阵，其中每一行都是C的一个特征向量。如果设P按照Λ中特征值的从大到小，将特征向量从上到下排列，则用P的前K行组成的矩阵乘以原始数据矩阵X，就得到了我们需要的降维后的数据矩阵Y。

至此我们完成了整个PCA的数学原理讨论。

# PCA算法步骤

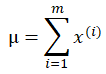


1、对所有数据特征进行中心化和归一化

对样本进行平移使其重心在原点，并且消除不同特征数值大小的影响，转换为统一量纲：

假设训练样本为一维变量：

样本均值为：



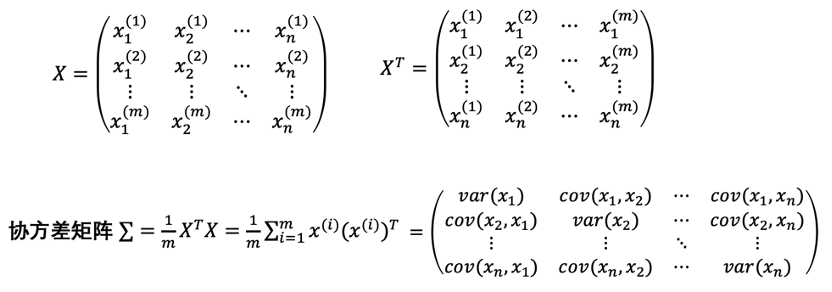
中心化：

归一化：

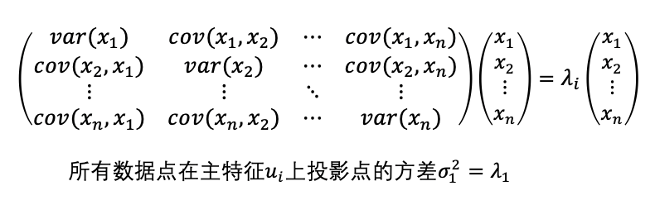


2、计算样本的协方差矩阵

协方差是对两个随机变量联合分布线性相关程度的一种度量；



3、对协方差矩阵求解特征值和特征向量



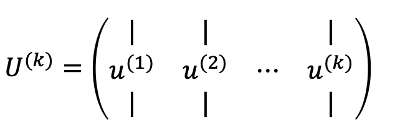
注意点：

1. 对称矩阵的特征向量相互正交，其点乘为0

② 数据点在特征向量上投影的方差，为对应的特征值，选择特征值大的特征向量，就是选择点投影方差大的方向，即是具有高信息量的主成分；次佳投影方向位于最佳投影方向的正交空间，是第二大特征值对应的特征向量，以此类推。

特征向量代表一组主成分空间的新坐标轴，特征值携带每个特征向量的方差数量的信息。因此，为了降低数据集的维度，我们准备选择那些具有较大方差的特征向量，丢弃那些具有较小方差的特征向量。

4、选取k个最大大特征值对应的特征向量，即是k个主成分



U是协方差矩阵所有的特征向量构成的矩阵，对应的特征值满足：λ1>λ2>⋯>λn，同时使其满足在主成分向量上投影的方差和占总方差的99%或者95%以上，即确定了k的选取。

|  |
| --- |
| PCA 不是从数据集选择某些特征而丢弃其他特征。PCA是在特征的组合基础上构造了一组新特征。从数学上讲，PCA 执行了一个线性转换，从原始特征集合转到由主成分组成的新空间。 |

# PCA应用

特征提取

简化运算

去除数据噪音

多维数据可视化

发现隐性相关变量。

# 实践

参考：

<https://github.com/jpegbert/MachineLearning/tree/master/jiangwei/pca>

既有纯python实现的，又有调用sklearn库实现的，还有针对降维后的分析过程

参考

<https://mp.weixin.qq.com/s/flJRWMPgu5GUCGTlP1fMWA>

<https://mp.weixin.qq.com/s/7Azblz_KgBsM03oE9_RWPw>

<https://mp.weixin.qq.com/s/R5_agyfByJRVvs5GsgO2Qg>

<https://mp.weixin.qq.com/s/9-nNNhhDWSYWy46u0hTazQ>

<https://mp.weixin.qq.com/s/uAlBtGTmtBSjcnp9bWQr5Q>

<https://mp.weixin.qq.com/s/WCR2LXLfNBMV9_KtjrTHuA>

<https://mp.weixin.qq.com/s/UoPhHm5Y5i4fK3syznO0vg>

<https://mp.weixin.qq.com/s/q8kXSQU92HDtyJfrSh1nPg>

<https://mp.weixin.qq.com/s/XqqTAvW86lTAw1A6uu-vtQ>