**所谓的特征值和特征向量**

在线性代数的最后，我们都会学矩阵的特征值分解，我们知道一个方阵A经过特征值分解后就得到特征向量和特征值了。那么，这个所谓的特征值和特征向量到底是什么东西呢？

我们一上来就会学到这样的一个公式：

Ax = λx，其中x是一个向量

这个式子是如此的简单粗暴，以致于从这个公式来看，给向量x乘上一个矩阵A，只是相当于给这个向量乘上了一个系数λ。**偌大一个矩阵A对向量x的作用竟然本质上不过只是和一个小小的数字λ相同而已！！！**

所以这个矩阵分解方法到底具有什么样的意义？

首先给出概念上的一种解释。所谓的特征值和特征向量，最重要的是理解“特征”这两个字，特征向量翻译为eigen vector, eigen这个单词来自德语，本义是在“本身固有的，本质的”。纯数学的定义下，并不能很明白地理解到底为什么叫做特征值和特征向量。但是举一个应用例子，可能就容易理解多了。

在图像处理中，有一种方法就是特征值分解。我们都知道图像其实就是一个像素值组成的矩阵，假设有一个100x100的图像，**对这个图像矩阵做特征值分解，其实是在提取这个图像中的特征，这些提取出来的特征是一个个的向量，即对应着特征向量。而这些特征在图像中到底有多重要，这个重要性则通过特征值来表示。**比如这个100x100的图像矩阵A分解之后，会得到一个100x100的特征向量组成的矩阵Q，以及一个100x100的只有对角线上的元素不为0的矩阵E，这个矩阵E对角线上的元素就是特征值，而且还是按照从大到小排列的（取模，对于单个数来说，其实就是取绝对值），也就是说这个图像A提取出来了100个特征，这100个特征的重要性由100个数字来表示，这100个数字存放在对角矩阵E中。在实际中我们发现，提取出来的这100个特征从他们的特征值大小来看，大部分只有前20（这个20不一定，有的是10，有的是30或者更多）个特征对应的特征值很大，后面的就都是接近0了，也就是说后面的那些特征对图像的贡献几乎可以忽略不计。我们知道，图像矩阵A特征值分解后可以得到矩阵Q和矩阵E：

那么反推出去，把右边的三个矩阵相乘肯定也能得到矩阵A。既然已经知道了矩阵E中只有前20个特征值比较重要，那么我们不妨试试把E中除了前20个后面的都置为0，即只取图像的前20个主要特征来恢复图像，剩下的全部舍弃，看看此时会发生什么：

原图：【注意：特征值分解要求必须是nxn的方阵，如果不是行列相等的方阵，请使用奇异值分解】



只取前10个特征值：



只取前20个特征值：



只取前50个特征值：



只取前100个特征值：



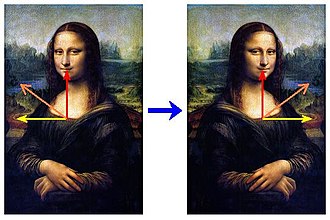
我们可以看到，在只取前20个特征值和特征向量对图像进行恢复的时候，基本上已经可以看到图像的大体轮廓了，而取到前50的时候，几乎已经和原图像无异了。明白了吧，这就是所谓的矩阵的特征向量和特征值的作用。

我们再来从数学定义上尝试去理解。对应一个给定的矩阵A，如果有一个向量v，使得矩阵A作用于v之后（即A和v相乘），得到的新向量和v仍然保持在同一直线上，像下面这样：

A\*v = lambda\*v

那么就称向量v是矩阵A的一个特征向量，而λ就是特征向量v对应的特征值【一个特征向量一定对应有一个特征值】。

注意这个定义中的要点，我们都知道矩阵其实就是一个线性变换，向量v在经过矩阵A这个线性变换之后，新向量和原来的向量v仍然保持在同一直线上，也就是说这个变换只是把向量v的长度进行了改变而保持方向不变（在特征值是负数的情况下，新向量的方向是原来方向的反向，即180°反方向）。

为了理解这个概念，我们再看维基百科上关于”特征向量”给出的一个《蒙娜丽莎》的例子：  


当蒙娜丽莎的图像左右翻转时，中间垂直的红色向量方向保持不变。而水平方向上黄色的向量的方向完全反转，因此它们都是左右翻转变换的特征向量。红色向量长度不变，其特征值为1。黄色向量长度也不变但方向变了，其特征值为-1。橙色向量在翻转后和原来的向量不在同一条直线上，因此不是特征向量。

所以归根结底，特征向量其实反应的是矩阵A本身固有的一些特征，本来一个矩阵就是一个线性变换，当把这个矩阵作用于一个向量的时候，通常情况绝大部分向量都会被这个矩阵A变换得“面目全非”，但是偏偏刚好存在这么一些向量，被矩阵A变换之后居然还能保持原来的样子，于是这些向量就可以作为矩阵的核心代表了。于是我们可以说：**一个变换（即一个矩阵）可以由其特征值和特征向量完全表述，这是因为从数学上看，这个矩阵所有的特征向量组成了这个向量空间的一组基底。而矩阵作为变换的本质其实不就把一个基底下的东西变换到另一个基底表示的空间中么？**