1.数据降维与SVD

3.SVD

3.1矩阵论预备知识

3.1.1特征值、特征向量

如果一个向量v是矩阵A的特征向量，将一定可以表示成下面的形式：

$$Av = \lambda v$$

其中，λ是特征向量v对应的特征值，一个矩阵的一组特征向量是一组正交向量。

思考：为什么一个向量和一个数相乘的效果与一个矩阵和一个向量相乘的效果是一样的呢？

答案：矩阵A与向量v相乘，本质上是对向量v进行了一次线性变换（旋转或拉伸），而该变换的效果为常数λ乘以向量v。

当我们求特征值与特征向量的时候，就是为了求矩阵A能使哪些向量（特征向量）只发生伸缩变换（**线性**），而变换的程度可以用特征值λ表示。

3.1.2特征值分解

对于矩阵A，有一组特征向量v，将这组向量进行正交化单位化，就能得到一组正交单位向量。**特征值分解**，就是将矩阵A分解为如下式：

$$A = Q\Sigma Q^{-1}$$

其中，Q是矩阵A的特征向量组成的矩阵，不同的特征值对应的特征向量线性无关。同时对于实对称矩阵而言，不同的特征向量必定正交。

Σ矩阵是一个对角阵，对角线上的元素就是特征值。

实例：

这里我们用一个简单的方阵来说明特征值分解的步骤。我们的方阵A定义为：

$$A=\begin{pmatrix}-1 & 1 & 0\\ -4 & 3 & 0\\ 1 & 0 & 2\end{pmatrix}$$

首先，由方阵A的特征方程，求出特征值。

$$|A-\lambda E|=\begin{vmatrix}-1-\lambda & 1 & 0\\ -4 & 3-\lambda 0 & \\ 1 & 0 & 2-\lambda\end{vmatrix}=(2-\lambda)\begin{vmatrix}-1-\lambda & 1\\ -4 & 3-\lambda\end{vmatrix}=(2-\lambda)(\lambda-1)^2=0$$

特征值为 $\lambda=2,1$ （重数是2）。

然后，把每个特征值λ带入线性方程组 $(A-\lambda E)x=0$ ，求出特征向量。

当λ=2时，解线性方程组 $(A-2E)x=0$ 。

$$(A-2E)=\begin{pmatrix}-3 & 1 & 0\\ -4 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}\rightarrow\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}$$

解得 $x_1=0,\ x_2=0$ 。特征向量为： $p_1=\begin{pmatrix}0\\ 0\\ 1\end{pmatrix}$ 。

当λ=1时，解线性方程组 $(A-E)x=0$

$$(A-2E)=\begin{pmatrix}-2 & 1 & 0\\ -4 & 2 & 0\\ 1 & 0 & 1\end{pmatrix}\rightarrow\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 2\\ 0 & 0 & 0\end{pmatrix}$$

$p_2=\begin{pmatrix}-1\\ -2$

$x_1 + x_3 = 0$， $x_2 + 2x_3 = 0$。特征向量为：$p_2 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$。

最后，方阵A的特征值分解为：

$$A = Q\Sigma Q^{-1} = \begin{pmatrix} 0 & -1 & -1 \\ 0 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & -1 \\ 0 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix}^{-1}$$

我们来分析一下特征值分解的式子，分解得到的Σ矩阵是一个对角矩阵，里面的特征值是由大到小排列的，**这些特征值所对应的特征向量就是描述这个矩阵变换方向**（从主要的变化到次要的变化排列）。

矩阵是高维的情况下，那么Σ矩阵就是高维空间下的一个线性变换，这个线性变换可能没法通过图片来表示，但是可以想象，这个变换也同样有很多的变化方向，**我们通过特征值分解得到的前N个特征向量，就对应了这个矩阵最主要的N个变化方向**。我们利用这前N个变化方向，就可以近似这个矩阵变换。也就是之前说的：**提取这个矩阵最重要的特征。**

总结：

特征值分解可以得到特征值与特征向量。

特征值表示的是这个特征到底有多么重要，而特征向量表示这个特征是什么，可以将每一个特征向量理解为一个线性的子空间，我们可以利用这些线性的子空间干很多事情。

不过，**特征值分解也有很多的局限，比如说变换的矩阵必须是方阵。当矩阵不是方阵的时候，这个时候就需要使用SVD对非方阵矩阵进行分解。**

3.2SVD分解

3.2.1思想

奇异值分解是一个能适用于**任意矩阵**的一种分解的方法，对于任意矩阵A总是存在一个奇异值分解：

$$A = U\Sigma V^T$$

假设A是一个m*n的矩阵。

那么得到的**U**是一个m*m的方阵，U里面的正交向量被称为**左奇异向量**。

Σ是一个m*n的矩阵，Σ除了对角线其它元素都为0，**对角线上的元素称为奇异值**。

VT是v的转置矩阵，是一个n*n的矩阵，它里面的正交向量被称为**右奇异值向量**。

由于U，V都是**正交阵**，可以得到:

$$U^TU = I, V^TV = I$$

而且一般来讲，我们会将Σ上的值按从大到小的顺序排列。上面矩阵的维度变化可以参照图4所示

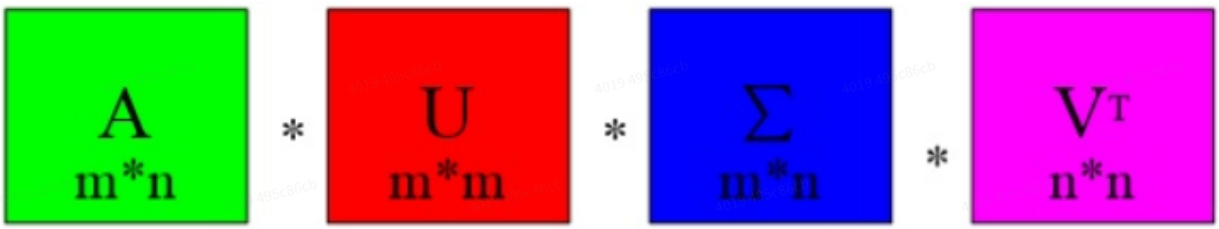

图4：奇异值分解中各个矩阵维度变化

3.2.2计算奇异值，奇异值向量

1.奇异值向量

把奇异值和特征值联系起来。**先构造出一个方阵A出来。**

首先，我们用矩阵A的转置乘以A，得到一个方阵，用这样的方阵进行特征分解，得到的特征值和特征向量满足下面的等式：

$$(A^TA)v_i = \lambda_i v_i$$

这里的v_i就是我们要求的**右奇异向量。（Why？？？）**

我们说 ATA的特征向量组成的矩阵就是我们SVD中的V矩阵(why？)

证明：

$$A = U\Sigma V^T \Rightarrow A^T = V\Sigma^T U^T \Rightarrow A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^2 V^T$$

所以ATA的特征向量就是我们要求的右奇异值向量。

同理，我们将A和A的转置做矩阵的乘法，得到一个方阵，用这样的方阵进行特征分解，得到的特征和特征向量满足下面的等式：

$$(AA^T)u_i = \lambda_i u_i$$

这里的u_i就是**左奇异向量。**

2.奇异值

奇异值求法有两种：

方法一：

$$A = U\Sigma V^T \Rightarrow AV = U\Sigma V^T V \Rightarrow AV = U\Sigma \Rightarrow Av_i = \sigma_i u_i \Rightarrow \sigma_i = \frac{Av_i}{u_i}$$

方法二：

通过下面可以看出：

$$A = U\Sigma V^T \Rightarrow A^T = V\Sigma^T U^T \Rightarrow A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^2 V^T$$

ATA的特征值矩阵等于奇异值矩阵的平方，（不能是AAT，维度不匹配）也就是说特征值和奇异值满足如下关系：

$$\sigma_i = \sqrt{\lambda_i}$$

3.2.3SVD的意义

思考：我们已经知道如何用奇异值分解任何矩阵了，那么问题又来了，一个m*n的矩阵A，你把它分解成m*m的矩阵U、m*n的矩阵Σ和n*n的矩阵V^T。。这三个矩阵中任何一个的维度似乎一点也不比A的维度小，而且还要做两次矩阵的乘法，这不是没事找事干嘛！把简单的事情搞复杂了么！并且我们知道矩阵乘法的时间复杂度为$O(n^3)$。O那奇异值分解到底要怎么做呢？

在奇异值分解矩阵中Σ里面的奇异值按从大到小的顺序排列，奇异值从大到小的顺序减小的特别快。**在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上。也就是说，剩下的90%甚至99%的奇异值几乎没有什么作用。**因此，我们可以**用前面r个大的奇异值来近似描述矩阵，**于是奇异值分解公式可以写成如下：

$$A_{m*n} \approx U_{m*r}\Sigma_{r*r}V^T_{r*n}$$

其中r是一个远远小于m和n的数，右边的三个矩阵相乘的结果将会使一个接近A的矩阵。如果r越接近于n，则相乘的结果越接近于A。如果r的取值远远小于n，从计算机内存的角度来说，右边三个矩阵的存储内存要远远小于矩阵A的。**所以在奇异值分解中r的取值很重要，就是在计算精度和时间空间之间做选择。**

3.3SVD分解的应用

3.3.1降维

通过奇异值分解的公式，我们可以很容易看出来，原来矩阵A的特征有n维。经过SVD分解后，可以**用前r个非零奇异值对应的奇异向量表示矩阵A**的主要特征，这样就把矩阵A进行了降维。

3.3.2压缩

通过奇异值分解的公式，我们可以看出来，矩阵A经过SVD分解后，**要表示原来的大矩阵A，我们只需要存储U、Σ、V三个较小的矩阵即可**。而这三个较小规模的矩阵占用内存上也是远远小于原有矩阵A的，这样SVD分解就起到了压缩的作用。