2.0.PCA(Principal Component Analysis)

1.PCA的思想

PCA的主要思想是将n维特征映射到k维上，这k维是全新的**正交特征**也被称为主成分，**是在原有n维特征的基础上重新构造出来的k维特征。**

1.1PCA做法

通过计算数据矩阵的协方差矩阵，然后得到**协方差矩阵的特征值特征向量**，选择**特征值最大**(即方差最大)的k个特征所对应的特征向量组成的矩阵。

这样就可以将数据矩阵转换到新的空间当中，实现数据特征的降维。

1.2做法解释

PCA的目标：

1）使得保留下来的 维度间 的相关性尽可能小。

2）使得保留下来的维度 含有尽可能多的原始信息（方差大）

所以，要知道各 维度间 的相关性 以及 各维度上的方差。那这个时候就想到了协方差矩阵。

协方差矩阵的含义：

主对角线上的元素是各个维度上的方差。

其他元素是两两维度间的协方差（即相关性）。

两者结合：

PCA目标的第一条反应到协方差矩阵中就是，使协方差矩阵中的非对角元素基本为0。

现在的目标是：Y= PX，

找到一个P，使Y的协方差矩阵（YY^T)变成一个对角矩阵。

$$
\begin{aligned}
D &= \frac{1}{m}YY^{\mathsf{T}} \\
&= \frac{1}{m}(PX)(PX)^{\mathsf{T}} \\
&= \frac{1}{m}PXX^{\mathsf{T}}P^{\mathsf{T}}
\end{aligned}
$$

$$= P(\frac{1}{m}XX^{\mathsf{T}})P^{\mathsf{T}}$$
$$= PCP^{\mathsf{T}}$$

我们要找的P不是别的，**而是能让原始协方差矩阵（XX^T）对角化的P。**所以后面就是对X的协方差矩阵分解即可得到最优的P，完成降维。

2.协方差矩阵

样本均值：

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

样本方差：

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

样本X和样本Y的协方差：

$$Cov(X,Y) = E[(X - E(X))(Y - E(Y))]$$
$$= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

由上面的公式，我们可以得到以下结论：

(1) **方差的计算公式是针对一维特征**，即针对同一特征不同样本的取值来进行计算得到；

而协方差则必须要求至少满足二维特征；方差是协方差的特殊情况。Cov(X,X)就是X的方差。

(2) 方差和协方差的除数是n-1,这是为了得到方差和协方差的无偏估计。

协方差为正时，说明X和Y是正相关关系；协方差为负时，说明X和Y是负相关关系；协方差为0时，说明X和Y是相互独立。

当样本是n维数据时，它们的协方差实际上是协方差矩阵(对称方阵)。例如，对于3维数据(x,y,z)，计算它的协方差就是：

$$Cov(X,Y,Z) = \begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

3.PCA算法两种实现方法

3.1基于特征值分解协方差矩阵实现PCA算法

3.1.1过程（记忆这个）

输入：数据集 $X = \{x_1, x_2, x_3, \ldots, x_n\}$，需要降到k维。

1) 去平均值(即去中心化)，即每一位特征减去各自的平均值。、

2) 计算协方差矩阵 $\frac{1}{n}XX^T$,注：这里除或不除样本数量n或n-1,其实对求出的特征向量没有影响。

3) 用特征值分解方法求协方差矩阵 $\frac{1}{n}XX^T$ 的特征值与特征向量。

4) 对特征值从大到小排序，选择其中最大的k个。然后将其对应的k个特征向量分别作为行向量组成特征向量矩阵P。

5) 将数据转换到k个特征向量构建的新空间中，即Y=PX。

3.1.2实例

以X为例，我们用PCA方法将这两行数据降到一行。

$$X = \begin{pmatrix} -1 & -1 & 0 & 2 & 0 \\ -2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

解：

1) 因为X矩阵的每行已经是零均值，所以不需要去平均值。

2) 求协方差矩阵：

$$C=\frac{1}{5}\begin{pmatrix}-1 & -1 & 0 & 2 & 0\\ -2 & 0 & 0 & 1 & 1\end{pmatrix}\begin{pmatrix}-1 & -2\\ -1 & 0\\ 0 & 0\\ 2 & 1\\ 0 & 1\end{pmatrix}=\begin{pmatrix}\frac{6}{5} & \frac{4}{5}\\ \frac{4}{5} & \frac{6}{5}\end{pmatrix}$$

3)求协方差矩阵的特征值与特征向量。

求解后的特征值为：

$$\lambda_1=2,\ \lambda_2=\frac{2}{5}$$

对应的特征向量为：

$$c_1\begin{pmatrix}1\\ 1\end{pmatrix},\ c_2\begin{pmatrix}-1\\ 1\end{pmatrix}$$

其中对应的特征向量分别是一个通解，C1和C2可以取任意实数。那么标准化后的特征向量为：

$$\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix},\ \begin{pmatrix}-\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}$$

4)矩阵P为：

$$P=\begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}$$

5)最后我们用P的第一行乘以数据矩阵X，就得到了降维后的表示：

$$Y=\begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}-1 & -1 & 0 & 2 & 0\\ -2 & 0 & 0 & 1 & 1\end{pmatrix}=\begin{pmatrix}-\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{pmatrix}$$

结果如图1所示：

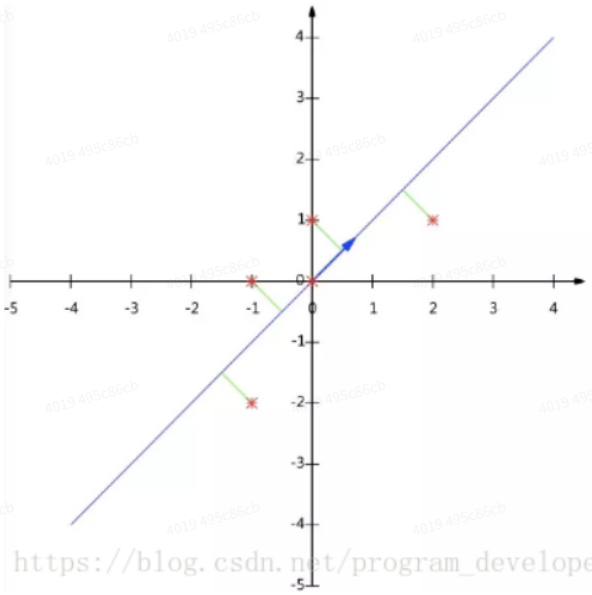


图1：数据矩阵X降维投影结果

4.选择降维后的维度K(主成分的个数)

如何选择主成分个数K呢？先来定义两个概念：

- average squared projection error：$\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2$，其中$x_{approx}^{(i)}$为映射值。
- total variation in the data：$\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2$

选择不同的K值，然后用下面的式子不断计算，选取能够满足下列式子条件的最小K值即可。

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}-x_{approx}^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2}\leq t$$

其中t值可以由自己定，比如t值取0.01，则代表了该PCA算法保留了99%的主要信息。当你觉得误差需要更小，你可以把t值设置的更小。

2.1KPCA

KPCA，中文名称”核主成分分析“，是对PCA算法的非线性扩展，言外之意，PCA是线性的，其对于非线性数据往往显得无能为力。**KPCA能够挖掘到数据集中蕴含的非线性信息。**

其实很好理解，就是原始的数据线性不可分，那么就升维变成线性可分。这个过程就需要使用核函数。升维后线性可分那么就采用PCA。

从 XX^T 操作变成对 $\phi(x)\phi(x^T)$

1.理论部分

1. 为了更好处理非线性数据，**引入非线性映射函数 $\phi(x)$ ，将原空间中的数据映射到高维空间。**

2. 引入了一个定理：**空间中的任一向量（哪怕是基向量），都可以由该空间中的所有样本线性表示。**

假设中心化后的样本集合X（d*N，N个样本，维数d维，样本”按列排列“），现将X映射到高维空间，得到 $\phi(x)$ ，**假设在这个高维空间中，本来在原空间中线性不可分的样本现在线性可分了，然后呢？想啥呢！果断上PCA啊！**

于是乎！假设D（D >> d）维向量 **Wi 为高维空间中的特征向量，为对应的特征值** λ_i ，高维空间中的PCA如下：

$$\Phi(X)\Phi(X)^T w_i = \lambda_i w_i \quad (1)$$

这个时候，在利用刚才的定理，将**特征向量Wi 利用样本集合 $\phi(x)$ 线性表示**，如下：

$$w_i = \sum_{k=1}^{N} \alpha_i \Phi(x_i) = \Phi(X)\alpha \quad (2)$$

然后，在把 $w_i\ (i = 1, \ldots, d)$ 代入上上公式，得到如下的形式：

$$\Phi(X)\Phi(X)^T \Phi(X)\alpha = \lambda_i \Phi(X)\alpha \quad (3)$$

进一步，等式两边同时左乘 $\Phi(X)^T$ ，得到如下公式：

$$\Phi(X)^T \Phi(X)\Phi(X)^T \Phi(X)\alpha = \lambda_i \Phi(X)^T \Phi(X)\alpha \quad (4)$$

这样做的目的是，构造两个$\Phi(X)^T$ $\Phi(X)$出来，进一步用核矩阵K（为对称矩阵）替代，其中：

$$\Phi(X)^T\Phi(X)\Phi(X)^T\Phi(X)\alpha=\lambda_i\Phi(X)^T\Phi(X)\alpha \tag{5}$$

于是，公式进一步变为如下形式：

$$K^2\alpha=\lambda_i K\alpha \tag{6}$$

两边同时去除K，得到了PCA相似度极高的求解公式：

$$K\alpha=\lambda_i\alpha \tag{7}$$

求解公式的含义就是**求K最大的几个特征值所对应的特征向量**，由于K为对称矩阵，所得的解向量彼此之间肯定是正交的。

但是，请注意，这里的α只是K的特征向量，但是其不是高维空间中的特征向量，回看公式（2），**高维空间中的特征向量w应该是由α进一步求出。**

2.核函数

(1)线性核函数(可视为特例)

$$K(x,x_i)=x\cdot x_i;$$

(2) p 阶多项式核函数

$$K(x,x_i)=[(x\cdot x_i)+1]^p;$$

(3)高斯径向基函数(RBF)核函数

$$K(x,x_i)=\exp\left(-\frac{\|x-x_i\|}{\sigma^2}\right);$$

(4)多层感知器(MLP)核函数

$$K(x,x_i)=\tanh[v(x\cdot x_i)+c];$$

2.2线性判别分析LDA

线性判别分析的基本思想是将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维数的效果，**投影后保证模式样本在新的子空间有 最大的类间距离 和 最小的类内距离，**即模式在该空间中有最佳的可分离性。

1.LDA假设以及符号说明：

假设对于一个 R^n 空间有m个样本分别为x1,x2,……xm 即 每个x是一个n行的矩阵，其中 n_i 表示属于i类的样本个数，假设有一个有c个类，则 $n_1+n_2+...+n_i+...+n_c=m$ 。

S_b …… 类间离散度矩阵

S_w …… 类内离散度矩阵

n_i …… 属于i类的样本个数

x_i …… 第i个样本

u …… 所有样本的均值

u_i …… 类i的样本均值

2.公式推导，算法形式化描述

根据符号说明可得类i的样本均值为：

$$\boldsymbol{u}_i=\frac{1}{\boldsymbol{n}_i}\sum_{x\in class\,i}\boldsymbol{x} \quad \cdots\cdots (1)$$

同理我们也可以得到总体样本均值：

$$\boldsymbol{u}=\frac{1}{\boldsymbol{m}}\sum_{i=1}^{m}\boldsymbol{x}_i \quad \cdots\cdots (2)$$

根据类间离散度矩阵和类内离散度矩阵定义，可以得到如下式子：

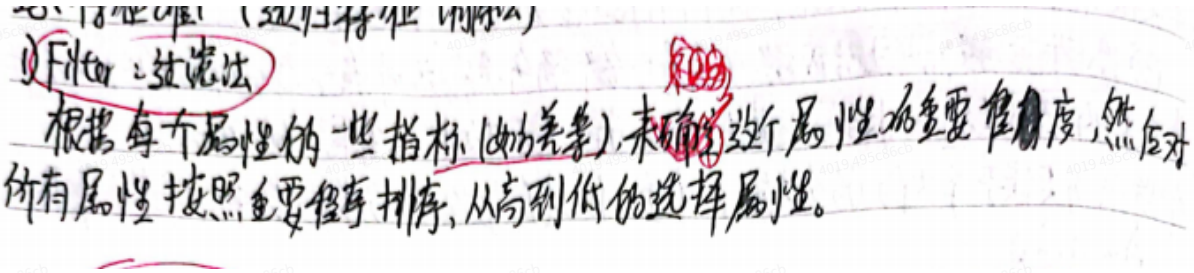
$$\mathbf{S}_b=\sum_{i=1}^{c}\boldsymbol{n}_i(\boldsymbol{u}_i-\boldsymbol{u})(\boldsymbol{u}_i-\boldsymbol{u})^T \quad \cdots\cdots (3)$$

$$\mathbf{S}_w=\sum_{i=1}^{c}\sum_{x_k\in class\,i}(\boldsymbol{u}_i-\boldsymbol{x}_k)(\boldsymbol{u}_i-\boldsymbol{x}_k)^T \quad \cdots\cdots (4)$$

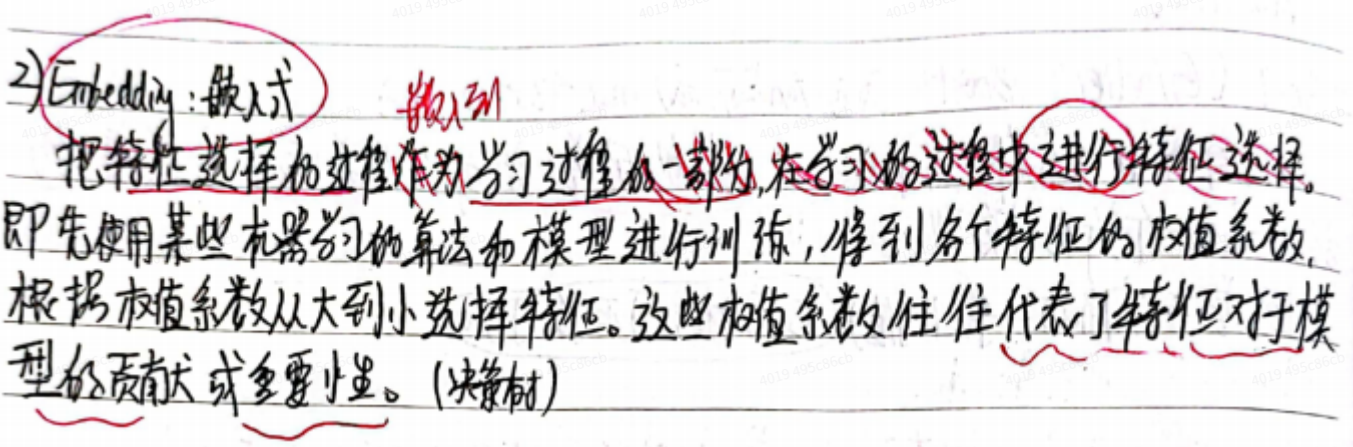
LDA做为一个分类的算法，我们当然希望它所分的**类之间耦合度低，类内的聚合度高**，即**类内离散度矩阵的中的数值要小，而类间离散度矩阵中的数值要大**，这样的分类的效果才好。

3.特征工程

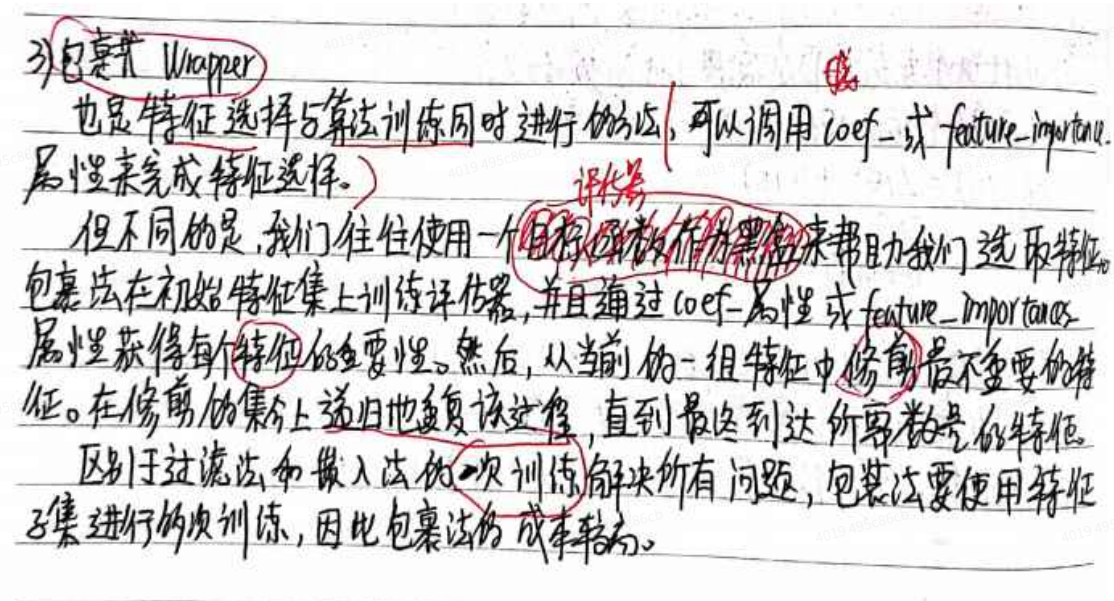
1.Filter:过滤法



2.Embedding:嵌入式



3.Wrapper:包裹式



最典型的方法：递归特征消除法（RFE）

它是一种贪心的优化算法，旨在找到性能最佳的特征子集。它反复创建模型，并在每次迭代时保留最佳特征或剔除最差特征，下一次迭代时，它会使用上一次建模中没有被选中的特征来构建下一个模型，直到所有特征都耗尽此。

然后，根据自己保留或剔除特征的顺序来对特征进行排名，

Date: \ \ Page:

最终选出一个最佳子集。

包裹法是所有特征选择方法中最有利于提升模型表现的，它可以使用很少的特征来达到很优秀的效果。但其计算量大，不太适合太大型的数据。

4.树模型对特征重要性进行评估

1. 随机森林（RF）简介

随机森林的算法可以用如下几个步骤概括：

1）用有抽样放回的方法（bootstrap）从样本集中选取n个样本作为一个训练集

2）用抽样得到的样本集生成一棵决策树。在生成的每一个结点：

2.1 随机不重复地选择d个特征

2.2 利用这d个特征分别对样本集进行划分，找到最佳的划分特征（可用基尼系数、增益率或者信息增益判别）

3）重复步骤1到步骤2共k次，**k即为随机森林中决策树的个数。**

4）用训练得到的随机森林对测试样本进行预测，并用票选法决定预测的结果。

2.特征重要性评估

计算每个特征在随机森林中的每颗树上做了多大的贡献，然后取个平均值，最后比一比特征之间的贡献大小。

好了，那么这个贡献是怎么一个说法呢？通常可以用基尼指数（Gini index）或者袋外数据（OOB）错误率作为评价指标来衡量。

2.1 基于基尼系数

我们将变量重要性评分（variable importance measures）用VIM来表示，将Gini指数用GI来表示，假设有J个特征X_1，X_2，X_3，…，X_J，I棵决策树，C个类别，现在要计算出每个特征X_j的Gini指数评分$VIM_j^{(Gini)}$，亦即第j个特征在RF所有决策树中节点分裂不纯度的平均改变量。

第i棵树节点q的Gini指数的计算公式为

$$GI_q^{(i)} = \sum_{c=1}^{|C|}\sum_{c'\neq c} p_{qc}^{(i)} p_{qc'}^{(i)} = 1 - \sum_{c=1}^{|C|}(p_{qc}^{(i)})^2 \tag{3-1}$$

Pqc的含义：节点q 中类别 c 所占的比例。

特征X_j在第i棵树节点q的重要性，即节点q分枝前后的Gini指数变化量为

$$VIM_{jq}^{(Gini)(i)} = GI_q^{(i)} - GI_l^{(i)} - GI_r^{(i)} \tag{3-2}$$

其中，$GI_l^{(i)}$和$GI_r^{(i)}$分别表示分枝后两个新节点的Gini指数。

如果，特征X_j在决策树i中出现的节点为集合Q，那么X_j在第i颗树的重要性为

$$VIM_j^{(Gini)(i)} = \sum_{q \in Q} VIM_{jq}^{(Gini)(i)} \tag{3-3}$$

假设RF 中共有I颗树，那么

$$VIM_j^{(Gini)} = \sum_{i=1}^{I} VIM_j^{(Gini)(i)} \tag{3-4}$$

最后，把所有求得的重要性评分做一个归一化处理即可。

$$VIM_j^{(Gini)} = \frac{VIM_j^{(Gini)}}{\sum_{j'=1}^{J} VIM_{j'}^{(Gini)}} \tag{3-5}$$

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

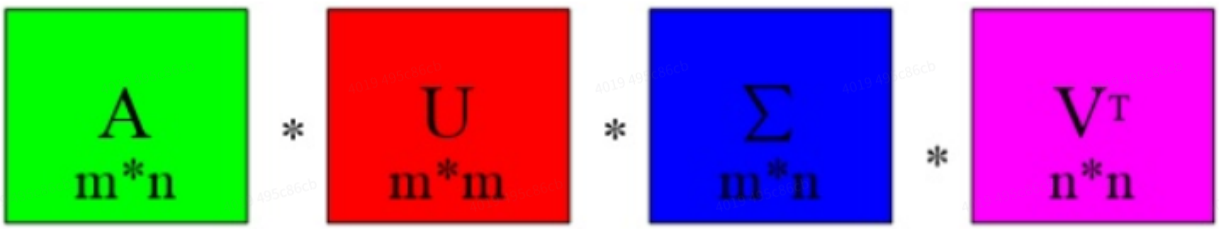


图4：奇异值分解中各个矩阵维度变化

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。