## 原理：

奇异分解与主成分分析法相似，任意一个M\*N的矩阵A（M行\*N列，M>N），可以被写成三个矩阵的乘积：

1. U：（M行M列的列正交矩阵）

2. S：（M\*N的对角线矩阵，矩阵元素非负），权重

3. V：（N\*N的正交矩阵的倒置）

即 A=U\*S\*V'（注意矩阵V需要倒置）

S是一个对角线矩阵，选取值较大的元素所在列，舍弃较小的元素，同时对U，V中的对应向量采取对应的舍弃，然后相乘法，即可得到原矩阵的近似，可以参考matlab中svds函数以理解。

博文：<http://www.cnblogs.com/LeftNotEasy/archive/2011/01/19/svd-and-applications.html>

<http://blog.sciencenet.cn/blog-696950-699432.html>

## 实例：

下面我们将用一个具体的例子展示svd的具体过程。

首先是A矩阵。

A =

5 5 0 5

5 0 3 4

3 4 0 3

0 0 5 3

5 4 4 5

5 4 5 5

（代表上图的评分矩阵）

使用matlab调用svd函数：

[U,S,Vtranspose]=svd(A)

U =

-0.4472 -0.5373 -0.0064 -0.5037 -0.3857 -0.3298

-0.3586 0.2461 0.8622 -0.1458 0.0780 0.2002

-0.2925 -0.4033 -0.2275 -0.1038 0.4360 0.7065

-0.2078 0.6700 -0.3951 -0.5888 0.0260 0.0667

-0.5099 0.0597 -0.1097 0.2869 0.5946 -0.5371

-0.5316 0.1887 -0.1914 0.5341 -0.5485 0.2429

S =

17.7139 0 0 0

0 6.3917 0 0

0 0 3.0980 0

0 0 0 1.3290

0 0 0 0

0 0 0 0

Vtranspose =

-0.5710 -0.2228 0.6749 0.4109

-0.4275 -0.5172 -0.6929 0.2637

-0.3846 0.8246 -0.2532 0.3286

-0.5859 0.0532 0.0140 -0.8085

分解矩阵之后我们首先需要明白S的意义。

可以看到S很特别，是个对角线矩阵。

每个元素非负，而且依次减小，具体要讲明白元素值的意思大概和线性代数的特征向量，特征值有关。

但是可以大致理解如下：

在线性空间里，每个向量代表一个方向。

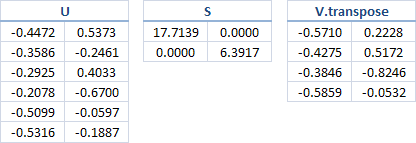
所以特征值是代表该矩阵向着该特征值对应的特征向量的方向的变化权重。

所以可以取S对角线上前k个元素。

当k=2时候即将S(6\*4)降维成S(2\*2)，

同时U(6\*6),Vtranspose(4\*4)相应地变为 U(6\*2),Vtranspose(4\*2).

如下图（图片里的usv矩阵元素值和我自己matlab算出的usv矩阵元素值有些正负不一致，但是本质是相同的）：



此时我们用降维后的U，S，V来相乘得到A2

A2=U(1:6,1:2)\*S(1:2,1:2)\*(V(1:4,1:2))' //matlab语句

A2 =

5.2885 5.1627 0.2149 4.4591

3.2768 1.9021 3.7400 3.8058

3.5324 3.5479 -0.1332 2.8984

1.1475 -0.6417 4.9472 2.3846

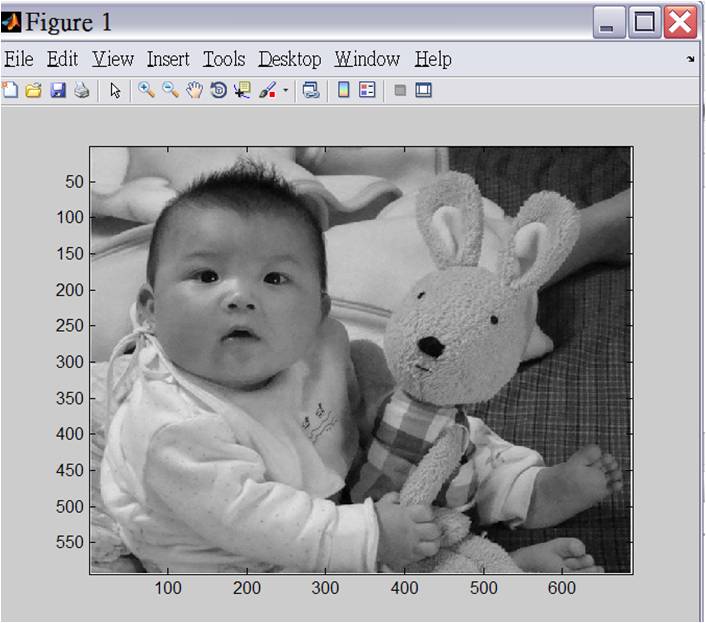
5.0727 3.6640 3.7887 5.3130

5.1086 3.4019 4.6166 5.5822

此时我们可以很直观地看出，A2和A很接近，这就是之前说的降维可以看成一种数据的有损压缩。

## 应用：

下面是一奇异分解在图像数据压缩上的应用：



原图

C=imread('graybaby.jpg');

k = input('請輸入 k 值：') %由使用者自行輸入奇異值個數

[U,S,V]=svds((double(C)),k);

B=U\*S\*V';

imagesc(B);

colormap(gray);



k=10恢复的图像