1 奇异值分解

奇异值分解将数据矩阵 $\mathbf{X}_{n\times p}$ （秩为 $r \le min(n,p)$）分解为

$$X_{n\times p} = U_{n\times r} D_{r\times r} V_{r\times p}^{\top}, \quad U^{\top}U = I_r, \quad V^{\top}V = I_r, \quad D = diag(\sqrt{\lambda_1},..., \sqrt{\lambda_r}), \tag{1}$$

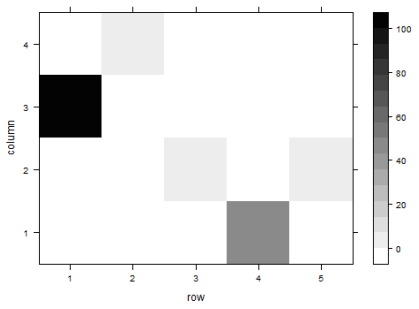
其中 $\lambda_1 \ge ... \ge \lambda_r > 0$ 为 $X^{\top}X$ 或 $XX^{\top}$ 的非 0 特征根, V 的第 k 个列向量是对应 λ_k 的 $X^{\top}X$ 的特征向量(刻画 X 的行向量)，U 的第 k 个列向量是 $XX^{\top}$ 对应 λ_k 的特征向量（刻画 X 的列向量）。R 中的奇异值分解函数 svd 的调用方式为：

```
svd( x, nu = min(n, p), nv = min(n, p) )
 # x: any n by p matrix
```

其中 x 是数据矩阵，svd 函数输出三部分: "d","u","v", 分别对应于公式 (1) 中的 D,U,V，选项中的 nu,nv 分别指定需要输出的 u,v 的列数（通常我们并不需要完整的 D,U,V），缺省值为 $\min(n,p)$。如果指定 nu $=r$, nv $=r$ (r=rank(x)), 就是公式 (1) 的情况，即完整的 SVD 分解。

例 1 (svd). 我们知道，奇异值分解的特征向量（U,V 的列）分别刻画矩阵的列向量特征和行向量特征，但具体情况则通常难以详细描述。对于下面的简单矩阵 A，我们考察其其奇异值分解。

```
> x=c(0,100,0,0,1,0,0,0,0,0,1,0,0,0,0,50,0,0,1,0)
> ( A=matrix(x,4,5) )
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    0    0    0
[2,]  100    0    0    0    0
[3,]    0    0    1    0    1
[4,]    0    0    0   50    0
## 用 levelplot 画格子点（不连续的）热图
library(lattice)
levelplot(A,col.regions = gray((10:1)/10) )
# 行列排列方式与我们看到的矩阵次序不同，改变行列和次序:
levelplot(t(A)[,4:1]  ,col.regions = gray((10:1)/10) )
```



A 是一个 4×5 矩阵，大部分元素为 0 或 1，有两个较大的元素 100，50。按列来看，100 位于第一列的 4×1 向量的第 2 个位置，50 位于第 4 个位置，我们认为列向量的第 2、4 位置是比较重要的。下面观察奇异值分解的特征向量:

```
> svd(A,nu=2,nv=2)
$d
[1] 100.000000  50.000000   1.414214   1.000000

$u
     [,1]  [,2]
[1,]    0    0
[2,]    1    0

$v
     [,1]  [,2]
[1,]    1    0
[2,]    0    0
```

```
[3,]    0    0                    [3,]    0    0
[4,]    0    1                    [4,]    0    1
                                  [5,]    0    0
```

A 的奇异值分解的第一个（最重要的）u,v 特征向量是

$$\mathbf{u}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$\mathbf{u}_1$ 第 2 个位置非 0，其它全为 0，说明 A 的列向量最重要的特征是 “第 2 位置最大，其它为 0 或较小”，这正描述了数字 100 在列向量中的位置。$\mathbf{v}_1$ 说明 A 的行向量最重要的特征是第 1 个位置，这正是数字 100 在行向量中的位置。因此 $\mathbf{u}_1, \mathbf{v}_1$ 分别刻画了矩阵 A 的最主要的列向量和行向量特征，两者一起决定了 “$(2,1)$ 元素是 A 的最重要的元素”。类似地，第 2 特征向量 $\mathbf{u}_2 = (0,0,0,1)^\top$, $\mathbf{v}_2 = (0,0,0,1,0)^\top$ 刻画了 A 的第二重要特征 (即 50 所在位置)。

练习 1. 截取 A 奇异值分解的前两阶，即 $A_2 = \sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1^\top + \sqrt{\lambda_2}\mathbf{u}_2\mathbf{v}_2^\top$, 画出 A_2 的热图.

练习 2（图像压缩）. 数据集 monalisa.txt 是一个 18x29 的黑白图像数据。

```
monalisa=read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/monalisa.txt")
monalisa=as.matrix( monalisa)
#  连续热图:   image()
image(monalisa, col=gray.colors(256),axes=F )
```

求数据集 monalisa 的 5 阶 SVD 逼近，画出热图，

例 2 (奇异值分解 SVD 与主成分分析 PCA). 当矩阵 X 是中心化矩阵时，PCA⇔SVD。假设中心化矩阵 X 的奇异值分解为 $X = UDV^\top$, 则 V 是 PCA 的载荷矩阵，$Y = XV = UD$ 为所有主成分 (y_{ik} 为第 i 个样本点的第 k 个主成分)。因此, 主成分分析也可由 svd 函数完成，PC 散点图的横轴为第一主成分，即 $XV = UD$ 的第一列；纵轴为第二主成分，即 UD 的第二列。利用奇异值分解做 PCA 与 R 的专用函数 princomp，prcomp 结果相同，后者会提供方差贡献率等内容，而奇异值分解不提供（但从给出的奇异值很容易计算得到）。奇异值分解做 PCA 的优势是计算效率较高（特别是对于大型数据），更高效的 SVD 函数还有：

```
rsvd (in package:rsvd),  fastsvd (in package: Bootsvd)
```

练习 3 （基因数据主成分分析）. 数据集

`http://staff.ustc.edu.cn/~ynyang/vector/data/genedata1.txt`

包含 $n = 800$ 个人的 $p = 1000$ 个基因的数据，第一列为每个人的种族（race: EUR 欧洲，EAS 东亚，AMR 美洲，SAS 南亚，AFR 非洲），2-1001 列为基因数据（取值 0,1,2）。应用 svd 函数求前两个主成分，画出其散点图，并把不同的种群 race 标记不同的颜色。各个种群在二维图上是否能较好地分开？（参见第 12 讲例 1）

例 3 (eigenface). SVD 在图像处理中的一个应用是所谓 eigenface（特征脸）的求解。该方法将每一个图像的像素灰度矩阵按列排成向量 (不再像练习 2 那样当作矩阵)，比如 92×112 像素的黑白图像排成长度为 10304(=92*112) 的向量。数据集 faces40.txt 是 40 个人的 92*112 图像数据，该数据有 40 行 (40 人), 10304 列 (代表 10304 个像素).

对该矩阵进行 SVD 分解，得到的 d,u,v 三部分, 其中 v 称为 eigenfaces, 是特征脸谱或代表性的脸谱（特征向量或基）。特征脸谱加权平均即可得到各个人脸图像的逼近。

```
> faces40 = read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/faces40.txt")
> dim(faces40)
[1]    40 10304
> faces40=as.matrix(faces40)
> face1 = faces40[1,]  # 第一个人的图像数据（向量）
> face1=matrix(face1, 92,112) # 向量还原成 92x112 像素矩阵
> image(face1,  col=gray((1:256)/256), axes=F)  # 画图

> svd(faces40)->faces40.svd
> d=faces40.svd[[1]] #40x1
> u=faces40.svd[[2]] #40x 40
> v=faces40.svd[[3]] #10304x 40

> eigenface1 =v[,1] # 第一个 eigenface/特征向量，长度为 10304
> eigenface1=matrix( eigenface1, 92,112) # 向量还原成 92x112 像素矩阵
> image(eigenface1, col=gray((1:256)/256), axes=F) # 第一个 eigenface
> image(matrix(v[,2],92,112), col=gray((1:256)/256), axes=F) # 第二个 eigenface
```

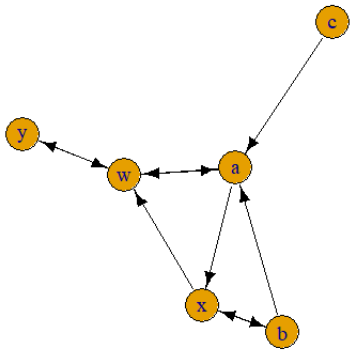
例 4 (网页排序) . 网页排序算法与奇异值分解有关。假设 n 个网页，定义所有网页的权威性 (authority) 指标为 $\mathbf{x}=(x_1,...,x_n)^\top$, 一个网页的权威性指标越大，越容易被别的网页链接。定义所有网页的中心性 (hub) 指标为 $\mathbf{y}=(y_1,...,y_n)^\top$, 一个网页的中心性指标越大，它越容易连接到其它网页。假设网页之间的邻接矩阵为 $n\times n$ 矩阵 A, 其 (i,j) 元 $a_{ij}=1$, 若网页 i 链接到网页 j, 否则 $a_{ij}=0$. HITS 模型假设

$$\mathbf{y}=cA\mathbf{x},\quad \mathbf{x}=dA^\top\mathbf{y}.$$

HITS 算法使用迭代方法，每次迭代时可以适当地将 $\mathbf{x},\mathbf{y}$ 单位化，直到迭代收敛:

```
x=x.ini=rep(1,n)  #x 初值
repeat{
```

```
 y=A%*%x;  y=y/sqrt(sum(y^2))
 x.new=t(A)%*%y; x.new=x.new/sqrt(sum(x.new^2))
 error=sum((x.new-x)^2)
 if (error<1e-5) break
 x=x.new
}
```



练习 4. 上图网络中两个网页之间的单箭头表示有链接（例如 b 连接到 a），双箭头表示相互有链接；(a) 写出邻接矩阵 A, 应用 HITS 迭代算法求各个网页的权威性指标并对网页排序; (b) 检查 HITS 算法收敛得到的权威性指标向量是否等于 $A^\top A$ 的最大特征向量（注意可能相差一个正负号）。

2 典则相关分析（CCA）

典则相关分析的 R 函数 cancor 只能处理样本 CCA，而不能处理仅有方差协方差矩阵的情况（即总体 CCA），cancor 也不计算每个样本的典则变量。下面我们使用自编函数 cca()。

```
cca(covmat=S,xlab=1:2) #总体cca, covmat为协方差矩阵
#xlab=1:2指定S的第一二行或列为x, 其它为y
cca(x,y) #样本cca, x,y: data frame
```

```
# 读入 cca 函数
source("http://staff.ustc.edu.cn/~ynyang/vector/lab/cca.txt")
## 第 18 讲例 1（总体 cca）
tmp=c(0.63, 0.24,0.06, -0.06, 0.07, 0.42)
s =matrix(0, 4,4)
s[lower.tri(s)]=tmp
s=s+t(s)
diag(s)=1
cca(covmat=s,xlab=1:2)
$cor.canonical
[1] 0.39 0.07 # 第一和第二典则相关系数

$xcoef  # x 的第一和第二典则变量的系数
        canon_v1  canon_v2
```

```
        1 -1.247798 0.3179603
        2  1.033039 0.7687192

$ycoef  # y 的第一和第二典则变量的系数
        canon_v1     canon_v2
        3 -1.1018763 -0.007089979
        4  0.4563537  1.002957091

#######
# 第 18 讲例 3:（样本 cca）

read.table("http://staff.ustc.edu.cn/~ynyang/vector/data/T9-12.DAT")->X
X=as.matrix(X)
colnames(X)=c("sale_growth", "sale_profit","sale_newacc","creativity","m_reasoning","a_reasoning","math")
apply(X,2, mean)->m
t(X)-m->X
X=t(X)
y=X[,1:3]
x=X[,4:7]

mycca=cca(x,y)
xcca=mycca$xcca # 三列分别是 50 个人的 x 的 3 个典则变量
ycca =mycca$ycca # 三列分别是 50 个人的 y 的 3 个典则变量
```

练习 5. 求出第 18 讲例 3 的三对典则变量，并画出散点图。

3 （选做）对应分析（CA：correspondence analysis）

例 6 (**对应分析**). 对应分析研究计数矩阵（称为列联表）的行与列之间的对应关系. 列联表各行代表某个属性变量的各个类别 (category)，各列代表第二个属性的各个类别。比如下表是考古数据的统计表，各行代表 6 个考古地点 (site1-6)，不同的地点可能代表不同的时期，各列代表 7 种燧石类型 (A-G)。

	a	b	c	d	e	f	g
1	20	3	4	42	18	0	13
2	85	3	12	0	0	0	0
3	26	40	8	0	0	26	0
4	20	1	4	13	58	0	4
5	67	10	23	0	0	0	0
6	26	29	8	3	0	33	1

我们感兴趣的问题包括

- 哪些考古地点类似？哪些不同？这是行之间比较的问题，但比较之前，需要将每一行归一化（除非各行总和相同），即每行的各个计数除以行的总数。
- 哪些燧石类型是类似的？这是列比较的问题，但比较之前需要列归一化，即每列的元素都出一该列的总数。
- 每个地点主要有哪些类型的燧石，或者每种燧石主要产出于那个考古地点？这是行列之间的关联问题。

R 软件包 FactoMineR 提供的 CA 函数用来对列联表数据进行对应分析。**R** 命令：

```
install.packages("FactoMineR")
library(FactoMineR)
myca=CA(X) #X: 列联表
```

对考古数据应用 CA 函数，得到下图所示的双标图可以看到 site1、4 相近（相似），D、G、E 相似. 而 site1、4 与 D、G、E 靠近，说明 site1,4 出土 DGE 较多。