第十五讲 奇异值分解

2024.4.29

上半学期：normal多元正态
下半学期：singular奇异值分解

奇异值分解(SVD)

列空间

对任一$n \times p$矩阵X，其列向量张成的空间（简称列空间）记作

$$C(X) = \{X\mathbf{t}: \mathbf{t} \in R^p\}$$

引理1.（1）列空间$C(X) = C(XX^\top)$；行空间$C(X^\top) = C(X^\top X)$.

（2）$X^\top X$与$XX^\top$有相同的非0特征根，特征向量相互关联。特别地，若$\mathbf{a}$是$X^\top X$的特征向量，则$\mathbf{b} = X\mathbf{a}$是$XX^\top$的特征向量；若$\mathbf{b}$是$XX^\top$的特征向量，则$\mathbf{a} = X^\top \mathbf{b}$ 是$X^\top X$的特征向量。

证: (1) $C(XX^\top) = \{XX^\top \mathbf{t}: \mathbf{t} \in R^p\} = \{X\mathbf{s}: \mathbf{s} = X^\top \mathbf{t} \in R^p\} \subset C(X)$

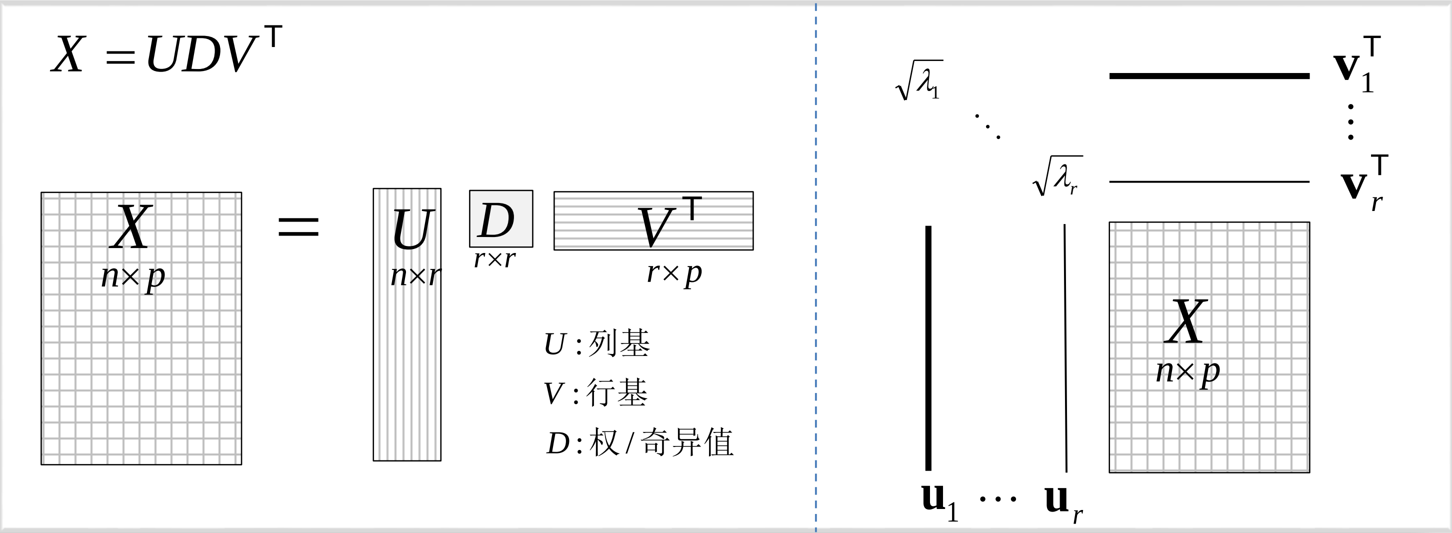
若$\mathbf{x} \in C(XX^\top)^\perp, XX^\top \mathbf{x} = 0 \Rightarrow \mathbf{x}^\top XX^\top \mathbf{x} = 0 \Rightarrow X^\top \mathbf{x} = 0 \Rightarrow \mathbf{x} \in C(X)^\perp$

$\Rightarrow C(XX^\top)^\perp \subset C(X)^\perp, C(X) \subset C(XX^\top)$.

(2). $X^\top \mathbf{b} = X^\top X\mathbf{a} = \mathbf{a}\lambda \Rightarrow XX^\top (X\mathbf{a}) = (X\mathbf{a})\lambda$.

引理1蕴含了奇异值分解（SVD, singular value decomposition）。特别地，为了刻画矩阵X的列空间特征，我们可以考虑$XX^\top$的列空间特征（特征向量）。对X的行也是如此。

SVD
$X = UDV^{\mathsf T}$



定理1(SVD). 任一秩为r的$n\times p$矩阵X可表示为

$$X = U_{n\times r}D_{r\times r}V^{\mathsf T}{}_{r\times p} = \sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1^{\mathsf T} + \sqrt{\lambda_2}\mathbf{u}_2\mathbf{v}_2^{\mathsf T} + \ldots + \sqrt{\lambda_r}\mathbf{u}_r\mathbf{v}_r^{\mathsf T}$$

其中$U^{\mathsf T}U = V^{\mathsf T}V = I_r$, $D = \mathrm{diag}(\sqrt{\lambda_1},\ldots,\sqrt{\lambda_r})$的对角元称为奇异值,
其中$\lambda_1 \ge \ldots \ge \lambda_r > 0$为$X^{\mathsf T}X$或$XX^{\mathsf T}$的$r$个正特征根，$U = (\mathbf{u}_1,\ldots,\mathbf{u}_r)$,
$V = (\mathbf{v}_1,\ldots,\mathbf{v}_r)$的各列分别是$XX^{\mathsf T}$，$X^{\mathsf T}X$的特征向量。

注：当X是中心化的数据矩阵，$Y = XV = UD$是主成分矩阵（第k列是第 k主成分）。

证明：$X^\top X$的非0特征根的特征方程：

$$X^\top X\mathbf{v}_i = \mathbf{v}_i\lambda_i, i = 1, \dots, r \ \Leftrightarrow X^\top XV = VD^2,$$

其中$D^2 = \mathrm{diag}(\lambda_1, \dots, \lambda_r)$

若X已经中心化, 则
$X^\top X = (n-1)S$

令 $Y = XV$

若X已经中心化，
则 Y 是主成分矩阵。

$$XX^\top XV = XVD^2 \Leftrightarrow XX^\top Y = YD^2，\ Y^\top Y = V^\top X^\top XV = D^2$$

Y的列是$XX^\top$的正交特征向量,模长分别为$\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r}$

令$U = YD^{-1}$ (Y单位化)

$U^\top U = I_r$，U的列是$XX^\top$的单位正交特征向量

综上 $Y = XV = UD$，左乘$V^\top \Rightarrow\ XVV^\top = UDV^\top$

因为$C(V) = C(X^\top X) = C(X^\top), VV^\top = P_V = P_{X^\top}$, 所以

$UDV^\top = XVV^\top = XP_V = XP_{X^\top} = X.$

SVD的一些评注

- 若X已经中心化，则 SVD $\Leftrightarrow$ PCA

  $X^\top X=(n-1)S$，$Y=XV=UD$ 是所有非0奇异值对应的主成分。

- 正交展开： $X=UDV^\top=\sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1^\top+\cdots+\sqrt{\lambda_r}\mathbf{u}_r\mathbf{v}_r^\top.$

- V、U分别是X的行、列特征： $XX^\top U=UD^2$，$X^\top XV=VD^2$

- 对偶方程：$XV=UD$，$X^\top U=VD$

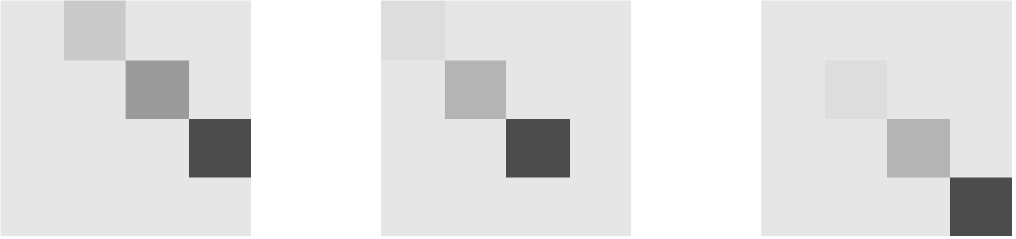
  对U，V的特定一列$\mathbf{u}$，$\mathbf{v}$ (对应的特征根λ)：$X\mathbf{v}=\mathbf{u}\sqrt{\lambda}$，$X^\top\mathbf{u}=\mathbf{v}\sqrt{\lambda}$

$$\begin{tikzcd} \mathbf{v}\in C(X^\top) \arrow[r, bend left, "X"] & \mathbf{u}\in C(X) \arrow[l, bend left, "X^\top"] \end{tikzcd}$$

虽然X不是方阵，没有特征向量，但$\{\mathbf{v}，\mathbf{u}\}$可看作是$X$及其伴随$X^\top$的不变量

例1.

$$X=\begin{pmatrix}0&1&0&0\\0&0&2&0\\0&0&0&3\\0&0&0&0\end{pmatrix},\quad XX^{\top}=\begin{pmatrix}1&0&0&0\\0&4&0&0\\0&0&9&0\\0&0&0&0\end{pmatrix},\quad X^{\top}X=\begin{pmatrix}0&0&0&0\\0&1&0&0\\0&0&4&0\\0&0&0&9\end{pmatrix},$$



$XX^{\top}$的特征根：$\lambda_1=9$，$\lambda_2=4$，$\lambda_3=1$。X的奇异值：3,2,1

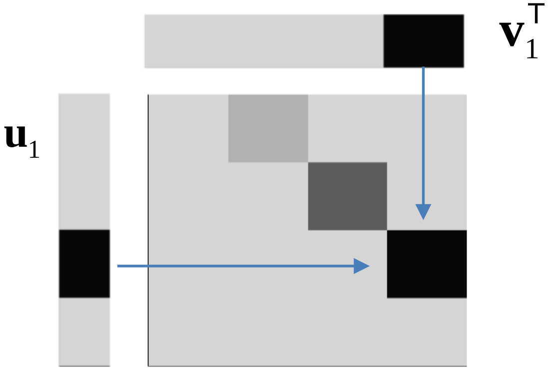
$$X=UDV^{\top}$$

其中$U=\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\\0&0&0\end{pmatrix}$，$V=\begin{pmatrix}0&0&0\\0&0&1\\0&1&0\\1&0&0\end{pmatrix}$，$D=\mathrm{diag}(3,2,1)$

$XX^\top$或$X^\top X$的最大特征根$\lambda_1=9$对应的特征向量

- $\mathbf{u}_1=(0,0,1,0)^\top$：$X$的第3行最重要，$(XX^\top)_{33}$最大
- $\mathbf{v}_1=(0,0,0,1)^\top$：$X$的第4列最重要，$(X^\top X)_{44}$最大

如下图，$\mathbf{u}_1\mathbf{v}_1{}^\top$定位矩阵$X$的最重要的位置：(3,4)



类似地，第二大特征值4对应的$\mathbf{u}_2=(0,1,0,0)^\top$，$\mathbf{v}_2=(0,0,1,0)^\top$，
$\mathbf{u}_2\mathbf{v}_2{}^\top$定位矩阵$A$的（2,3）位置，$\mathbf{u}_3\mathbf{v}_3{}^\top$定位矩阵$A$的（1,2）

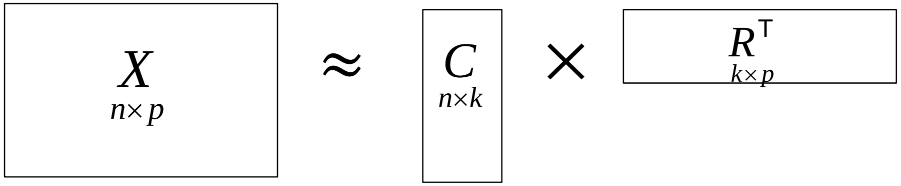
1阶逼近： $$3\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

2阶逼近： $$3\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} + 2\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

3阶即为原数据： $$3\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} + 2\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}} + \mathbf{u}_3\mathbf{v}_3^{\mathrm{T}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix} = X$$

矩阵分解和低秩逼近

矩阵的低秩逼近(low rank approximation)或矩阵分解逼近(matrix factorization)可以将数据矩阵在低维（低秩）空间上表示，利于发现数据结构或压缩数据。



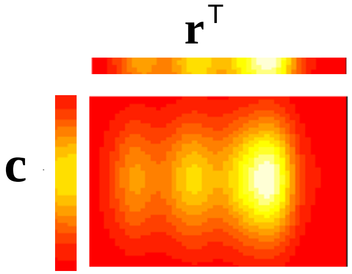
$$\underset{n\times p}{X} \approx \underset{n\times k}{C} \times \underset{k\times p}{R^{\mathsf{T}}}$$

$$k < r = rank(X)$$

矩阵分解不唯一，我们感兴趣的是低秩(小k)逼近或者对C，R有额外的限制的情形。我们将看到，在平方误差意义下，最优的秩k逼近是奇异值分解的前k项,这说明了奇异值分解的优良性

首先注意到，任意一个秩为1的矩阵$X_{n\times p}$可以表示为两个向量的外积：

$$\underset{n\times p}{X} = \underset{n\times 1}{\mathbf{c}}\ \underset{p\times 1}{\mathbf{r}}^{\mathsf{T}},$$



- 对任何矩阵$X=(x_{ij})$，我们以秩1矩阵逼近X：

$$\underset{n\times p}{X} = \underset{n\times 1}{\mathbf{c}}\ \underset{m\times 1}{\mathbf{r}}^{\mathsf{T}} + E,\quad E=(\varepsilon_{ij})\text{误差}$$

$$\Leftrightarrow \text{乘积模型：}\ x_{ij}=c_i r_j+\varepsilon_{ij}$$

- 我们也可以应用方差分析中的“可加模型”

$$x_{ij}=c_i+r_j+\varepsilon_{ij} \ \Leftrightarrow\ \underset{n\times p}{X} = \underset{n\times 1}{\mathbf{c}}\ \underset{m\times 1}{\mathbf{1}}^{\mathsf{T}} + \underset{n\times 1}{\mathbf{1}}\ \underset{m\times 1}{\mathbf{r}}^{\mathsf{T}} + E$$

近似逼近矩阵X,其中$\mathbf{r}$，$\mathbf{c}$未知，可称为行效应和列效应。

可加模型不易推广到更高秩情形。

矩阵分解

引理2. 任一秩k矩阵$X_{n\times p}$可表示为矩阵分解形式$X=CR^{\top}$,其中C, R分别是$n\times k$，$p\times k$列满秩矩阵。

证：取$\mathbf{t}_1,\dots,\mathbf{t}_k$为行空间C($X^{\top}$)的一组基，$R=(\mathbf{t}_1,\dots,\mathbf{t}_k)$,
记X的第i行为$\mathbf{x}_i^{\top}$，存在$\mathbf{s}_i$,使得$\mathbf{x}_i=R\mathbf{s}_i$，$i=1,\dots,n$,所以
$X=(\mathbf{x}_1,\dots,\mathbf{x}_n)^{\top}=(R\mathbf{s}_1,\dots,R\mathbf{s}_n)^{\top}=CR^{\top}$。

$$C_{n\times k}=(\mathbf{c}_1,\dots,\mathbf{c}_k)=(\mathbf{s}_1,\dots,\mathbf{s}_n)^{\top},$$
$$R_{p\times k}=(\mathbf{r}_1,\dots,\mathbf{r}_k)=(\mathbf{t}_1,\dots,\mathbf{t}_n)^{\top}$$

两种理解

- 正交展开：$X=CR^{\top}=\mathbf{c}_1\mathbf{r}_1^{\top}+\cdots+\mathbf{c}_k\mathbf{r}_k^{\top}$
  C的列向量$\{\mathbf{c}_i,i=1,\dots,k\}$ 是X的列空间的基;
  R的列向量$\{\mathbf{r}_i,i=1,\dots,k\}$是$X$的行空间的基。

列基
$X = C\ \ R^{\top}$
行基

- 行列关系：$X=CR^{\top}=\left(\boldsymbol{s}_i^{\top}\mathbf{t}_j\right)$:　$x_{ij}=\boldsymbol{s}_i^{\top}\mathbf{t}_j$
  $\{\mathbf{s}_i,i=1,\dots,n\}$是$X$的各行(样本)的某种feature;
  $\{\mathbf{t}_j,j=1,\dots,p\}$是$X$的各列(变量)的某种feature.
  （参见例2）

当$\mathbf{s}_i$与$\mathbf{t}_j$相似，
$x_{ij}=\boldsymbol{s}_i^{\top}\mathbf{t}_j$较大

例2（推荐系统的矩阵分解算法）. 2006年*Netflix*推荐数据比赛优胜者使用了矩阵分解算法。用户$U1$，$U2$,....给电影$M1, M2$,...打分(1-5分), 假设用户个数n,电影个数p,用户i给电影j的打分为$x_{ij}, X_{n\times p}=(x_{ij})$。

每个观众只看过少数电影，所以该矩阵大部分元素为空。我们的任务是填充上这些空格，进行推荐。

	M1	M2	M3	M4
U1	5	3	-	1
U2	4	-	-	1
U3	1	1	-	5
U4	1	-	-	4
U5	-	1	5	4

来源：http://www.albertauyeung.com/post/python-matrix-factorization/

经典算法：协同过滤

协同过滤算法 (collaborative filtering) 假设对某些电影评价类似/相反的用户对其它电影也应有类似/相反的评价，是推荐系统中最常见的假设。

	M1	M2	M3	M4
U1	5	3	?	1
U2	4	?		1
U3	1	1		5
U4	1			4
U5		1	5	4

U2对M2的评价？
U1，U2都评价了M1和M4，打分类似。U1对M2的评价为3，我们推测U2对M2的评价大概为3

U1对M3的评价？
U1似乎与U3，4，5相反，U5的评价为5，故推测U1对M3的评价较小，比如1.

	M1	M2	M3	M4
U1	5	3	1	1
U2	4	2	1	1
U3	1	1	5	5
U4	1	1	4	4
U5	1	1	5	4

Netflix算法：矩阵分解

假设一个人对某个电影评价高，是因为该电影具有的某些特征恰好是该观众/用户喜欢的类型。假设电影的特征(潜变量)有k个，比如题材、导演等等。这些特征在电影和用户上的体现：

- $\mathbf{s}_i = \left(s_{i1},\ldots,s_{ik} \right)^{\mathsf{T}}$，$s_{if}$ =用户i的对特征 f 的喜好。
- $\mathbf{t}_j = \left(t_{j1},\ldots,t_{jk} \right)^{\mathsf{T}}$，$t_{jf}$ =电影 j 的 特征 f 刻画。

假设模型：$\mathbf{s}_i$和$\mathbf{t}_j$的相似程度决定了用户i给电影 j的打分：

用户i给电影j的打分：$x_{ij} \approx \mathbf{s}_i^{\mathsf{T}} \mathbf{t}_j = \sum_{f=1}^{k} s_{if} t_{jf}$

$$= \sum_{f=1}^{k} (\text{用户}\, i \text{ 对 特征} f \text{ 的喜好程度}) \times (\text{电影 } j \text{ 含有特征 } f \text{ 的程度})$$

现假设有n个用户，p个电影，用户i给电影j的打分为x_{ij}。

求解$\left\{ \mathbf{s}_i, \mathbf{t}_j,\ 1 \le i \le n,\ 1 \le j \le p \right\}$，最小化误差平方和

$$J = \sum_{i,j} \left(x_{ij} - \mathbf{s}_i^{\mathsf{T}} \mathbf{t}_j \right)^2 = \| X - CR^{\mathsf{T}} \|_F^2 \ .$$

- 如果X没有空值，我们将会看到，最优解为SVD的前k阶展开；
- Netflix：如果有空值，误差平方和只对非空格子求和，使用梯度下降法求解；
- 如果有空值，也可先使用协同过滤填充，再使用SVD重写拟合，调整填充内容。

---

*Netflix*矩阵分解推荐算法(matrix factorization，Stochastic gradient descent)

目标：假设 $J(\theta)$可写为$\sum J_k(\theta_k)$，最小化$J(\theta)$

梯度(Gradient)：$\nabla J_k(\theta_k) = \dfrac{\partial J_k(\theta_k)}{\partial \theta_k}$

算法：$\theta_k^{(\text{update})} \leftarrow \theta_k^{(0)} - \eta \nabla J(\theta_k^{(0)})$

SVD及矩阵的最佳低秩逼近

低秩逼近问题

给定$n\times p$矩阵X,求解秩为k的$n\times p$矩阵A，使得误差平方和最小:

$$\min_{rank(A)=k} \| X - A \|_F^2 = \min_{\substack{C_{n\times k}, R_{p\times k}\\ \text{列满秩}}} \| X - CR^{\mathrm{T}} \|_F^2$$

其中矩阵的Frobenius 模定义为$\| A \|_F = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{tr(A^{\mathrm{T}} A)}.$

❑ 上述低秩逼近的最优解是k-阶SVD（下页定理2）

❑ 对特定的问题，矩阵分解施加一些限制将会得到不同于SVD的其它方法。例如

- ❖ 如果限制R的元素为0，1，代表k个类别的哑变量表示，得到的解将会把数据点聚集为k类（聚类分析）。
- ❖ 若X元素非负，要求C，R元素也是非负，那么得到非负矩阵分解（non-negative matrix factorizaton，NMF），该方法因为容易解释且具有聚类效果而颇为流行。

SVD的最优性

定理2（Eckart - Young - Mirsky）.

若$X_{n\times p}$的奇异值分解为$X=UDV^{\top}=\sum_{i=1}^{r}\sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^{\top}$， $r=rank(X)$,则对任何$k\le r$,

$$\min_{\substack{A\in R^{n\times p}\\ rank(A)=k}} \| X-A\|_F^2=\| X-\sum_{i=1}^{k}\sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^{\top}\|_F^2=\sum_{i=k+1}^{r}\lambda_i$$

即X的最优秩k逼近为$A_{\mathrm{opt}}=\sum_{i=1}^{k}\sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i^{\top}=U_kD_kV_k^{\top}$，其中$U_k=(\mathbf{u}_1,\dots,\mathbf{u}_{\mathrm{k}})$,

$V_k=(\mathbf{v}_1,\dots,\mathbf{v}_{\mathrm{k}}), D_k=diag(\sqrt{\lambda_1},\dots,\sqrt{\lambda_k})$。

证明：首先,对任何给定的$p\times k$矩阵R,不妨假设R是列正交且单位化的，即$R^{\top}R=I_k$,则必定有

$$\min_{C\in R^{n\times k}} \| X-CR^{\top}\|_F^2=\| X-XRR^{\top}\|_F^2 \qquad （*）$$

如果R不是列正交的，那么我们可以把R的列向量Schmidt正交化：$R=\tilde{R}T,\tilde{R}^{\top}\tilde{R}=I_k$,

T是上三角，则$CR^{\top}=CT^{\top}\tilde{R}^{\top}\overset{\Delta}{=}\tilde{C}\tilde{R}^{\top}$

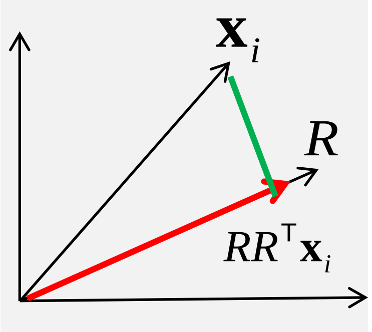
这是因为对于给定的R，$\| X-CR^{\top}\|_F^2=\sum_{i=1}^{n}\|\mathbf{x}_i-R\mathbf{c}_i\|^2$,其中

$X=(\mathbf{x}_1,\dots,\mathbf{x}_n)^{\top},C=(\mathbf{c}_1,\dots,\mathbf{c}_n)^{\top}$。

由最小二乘性质，每一项$\| \mathbf{x}_i - R\mathbf{c}_i \|^2$ 在 $R\mathbf{c}_i = P_R\mathbf{x}_i$

$= R\left(R^\top R\right)^{-1} R^\top \mathbf{x}_i = RR^\top \mathbf{x}_i$，即$\mathbf{c}_i = R^\top \mathbf{x}_i, C = XR$时达到最小。

$$\Rightarrow \min \sum_{i=1}^{n} \| \mathbf{x}_i - R\mathbf{c}_i \|^2 = \sum_{i=1}^{n} \| \mathbf{x}_i - RR^\top \mathbf{x}_i \|^2 = \| X - XRR^\top \|_F^2 。$$



其次，注意到:

$$\begin{aligned} \| X - XRR^\top \|_F^2 &= \mathrm{tr}\left(X - XRR^\top\right)^\top \left(X - XRR^\top\right) \\ &= \mathrm{tr}\left(X^\top X - X^\top XRR^\top - RR^\top X^\top X + RR^\top X^\top XRR^\top\right) \\ &= \mathrm{tr}(X^\top X) - \mathrm{tr}(R^\top X^\top XR) = \| X \|_F^2 - \| XR \|_F^2 \end{aligned}$$

因此,

$$\min \| X - XRR^\top \|_F^2 \Leftrightarrow \max \quad \| XRR^\top \|_F^2 = \max \| XR \|_F^2 = \max tr(R^\top X^\top XR)$$

记R的各列$R=(\mathbf{r}_1,\dots,\mathbf{r}_k)$,则 $R^\top X^\top XR=\left(\mathbf{r}_i{}^\top X^\top X\mathbf{r}_j{}^\top, 1\le i, j\le k\right)$,所以

$$\|XR\|_F^2=tr(R^\top X^\top XR)=\mathbf{r}_1{}^\top X^\top X\mathbf{r}_1+\dots+\mathbf{r}_k{}^\top X^\top X\mathbf{r}_k$$

因为$R^\top R=I_k$，$\mathbf{r}_1,\dots,\mathbf{r}_k$是相互正交的单位长向量。

我们只需极大化各个加项(R的各个方向上的投影长度):

$$\max_{\|\mathbf{r}_1\|=1}\mathbf{r}_1{}^\top X^\top X\mathbf{r}_1,\ \dots,\ \max_{\|\mathbf{r}_k\|=1,\mathbf{r}_k\perp\mathbf{r}_1,\dots,\mathbf{r}_{k-1}}\mathbf{r}_k{}^\top X^\top X\mathbf{r}_k \qquad (**)$$

注意如果X已经中心化，
$X^\top X=(n-1)S$,
$(**)$ 是PCA的极大化问题。

上述各项的最大值分别为$\lambda_1,\dots,\lambda_k$,最大值在$X^\top X$的前$k$个特征向量

$\mathbf{v}_1,\dots,\mathbf{v}_k$达到，即最优$\hat{R}=V_k=(\mathbf{v}_1,\dots,\mathbf{v}_k)$. 所以

$$\max\|X\hat{R}\|_F^2=\|XV_k\|_F^2=\lambda_1+\dots+\lambda_k.$$

$$\min\|X-XRR^\top\|_F^2 \quad =\|X-XV_kV_k{}^\top\|_F^2=tr(X^\top X)-\|XV_k\|_F^2=\lambda_{k+1}+\dots+\lambda_r$$

因为$XV_k=U_kD_k$，所以X的最佳逼近为

$$X\hat{R}\hat{R}^\top=XV_kV_k{}^\top=U_kD_kV_k{}^\top=\sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1{}^\top+\dots+\sqrt{\lambda_k}\mathbf{u}_k\mathbf{v}_k\text{。}$$

中心化矩阵的SVD等价于PCA

如果 X 已经中心化，我们在第1-2页已经看到PCA与SVD的等价性。
从优化的角度来看，从定理2的证明过程，求解低秩逼近等价于
极大化方差：

低秩逼近 $\min \| X - XRR^{\top} \|_F^2 \Leftrightarrow \max \| XR \|_F^2$

$$\Leftrightarrow \max_{\mathbf{r}_1,\ldots,\mathbf{r}_k\text{正交，模长}1} \left(\mathbf{r}_1^{\top} X^{\top} X \mathbf{r}_1 + \ldots + \mathbf{r}_k^{\top} X^{\top} X \mathbf{r}_k \right)$$

$\Leftrightarrow$ 极大化加和中的每一项：

$$\max_{\|\mathbf{r}_1\|=1} \left(\mathbf{r}_1^{\top} X^{\top} X \mathbf{r}_1 \right), \quad \max_{\mathbf{r}_2 \perp \mathbf{r}_1, \|\mathbf{r}_2\|=1} \left(\mathbf{r}_2^{\top} X^{\top} X \mathbf{r}_2 \right), \ \ldots\ , \quad \max_{\mathbf{r}_k \perp (\mathbf{r}_1,\ldots,\mathbf{r}_{k-1}), \|\mathbf{r}_k\|=1} \left(\mathbf{r}_k^{\top} X^{\top} X \mathbf{r}_k \right)$$

这是PCA极大化投影方差问题。

总之，中心化X的SVD等价于PCA，若

$$X = UDV^{\top}$$

则V的各列是主成分方向，$Y = XV = UD$的第k列是第k主成分。

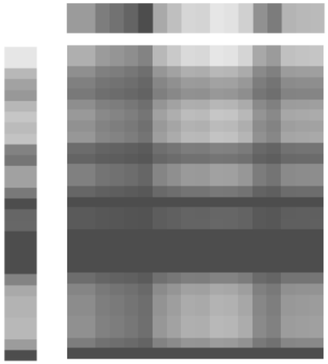
SVD应用1：图像压缩

图像压缩中很少采用SVD逼近方法，我们这里采用SVD压缩图像数据知识为了演示目的。

例3. 下表29 × 18 矩阵是头像数据，元素为 0-1，1：白，0：黑。

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1
1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1
1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

$1: \sqrt{\lambda_1}\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}}$



轮廓

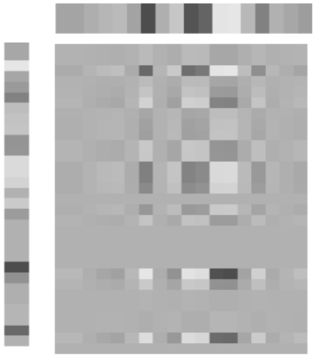
$2: \sqrt{\lambda_2}\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}}$

脸部水平特征：
头发,眼睛,嘴巴。
竖直方向边际

$3: \sqrt{\lambda_3}\mathbf{u}_3\mathbf{v}_3^{\mathrm{T}}$

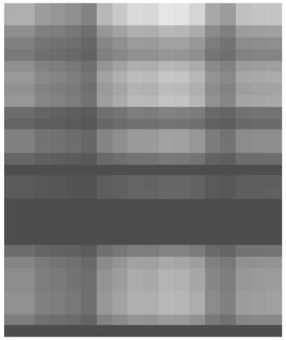
脸部垂直轮廓

$4: \sqrt{\lambda_4}\mathbf{u}_4\mathbf{v}_4^{\mathrm{T}}$

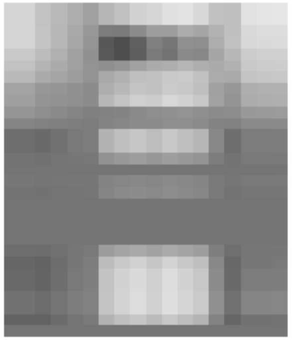


鼻子

1



1+2



1+2+3

1+2+3+4

SVD应用2：网页搜索排序

来源：Gilbert Strang (2016) **Introduction to Linear Algebra**, 5th ed

网页排序算法HITS与谷歌的PageRank都出现在1998-1999年，方法也有类似之处，都与SVD有关。

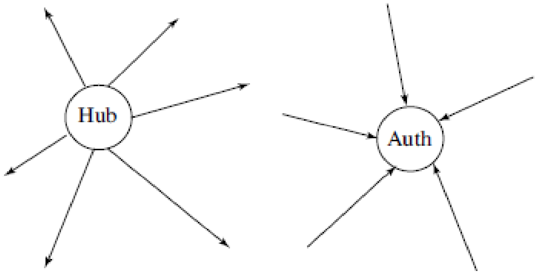
邻接矩阵

网页之间以超链接建立联系。定义邻接矩阵(adjacency matrix)

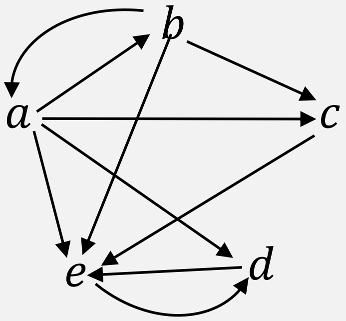
$$A=(a_{ij}), \qquad a_{ij}=\begin{cases}1 & \text{若}i\text{链接到}j\ (\text{记作}i\to j)\\ 0 & \text{否则}\end{cases}.$$

Hub, Authority

链入(in-link): 一个网页被多个网页链入，则其权威性Authority较大。
链出(out-link): 一个网页如果连接到多个其它网页，则其hub较大。



例4. 5个网页链接情况如图。网页a指向其它所有4个网页，是链出最多的网页；网页e被4个网页链接，是链入最多的网页。



link-in

Link-out

$$A = \begin{array}{c|ccccc|c} & a & b & c & d & e & 合计 \\ a & 0 & 1 & 1 & 1 & 1 & 4 \\ b & 1 & 0 & 1 & 0 & 1 & 3 \\ c & 0 & 0 & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 0 & 0 & 1 & 1 \\ e & 0 & 0 & 0 & 1 & 0 & 1 \\ 合计 & 1 & 1 & 2 & 2 & 4 & \end{array}$$

如果以网页链入个数（矩阵的列和）作为authority重要性度量，则在这种度量下，e最受欢迎（4分）。网页c和d的得分都是2，但网页d被权威性较高的e链入，故d的authority应该大于c。

所以在计算一个网页的权威性计分的时候，不但要考虑链入个数，也要考虑链入网页的权威性。

HITS算法

HITS算法的基本思想：一个权威性网页应该有重要的*hub*网页链接指向它,一个重要的*hub*网页应该链接指向一些权威网页。

假设所有网页的权威性计分为向量$\mathbf{v}$,*hub*重要性计分为向量$\mathbf{u}$。

假设网页之间的邻接矩阵为A（行为$\mathbf{a}_i$，列为$\boldsymbol{\alpha}_j$）。

假设网页j的权威性v_j与指向它的网页的hub计分总值成正比:

$$v_j = c\sum_{k \to j} u_k = c\boldsymbol{\alpha}_j^{\mathsf{T}}\mathbf{u}, \quad j = 1,2,\ldots,n \quad \Leftrightarrow \quad \mathbf{v} = cA^{\mathsf{T}}\mathbf{u}.$$

网页i的hub计分u_i依赖于它指向的网页的权威性总分:

$$u_i = d\sum_{i \to j} v_j = c\mathbf{a}_i^{\mathsf{T}}\mathbf{v}, \quad i = 1,2,\ldots.,n \quad \Leftrightarrow \quad \mathbf{u} = dA\mathbf{v}.$$

注意$\mathbf{u} = cA\mathbf{v}$和$\mathbf{v} = dA^{\mathsf{T}}\mathbf{u}$正是SVD的对偶方程。

$\Rightarrow \mathbf{u} = cA\ \mathbf{v} = cdAA^{\mathsf{T}}\mathbf{u}$， $\mathbf{v} = dA^{\mathsf{T}}\mathbf{yu} = cdA^{\mathsf{T}}A\mathbf{v}$

$\mathbf{u}$是AA^{T}的特征向量，$\mathbf{v}$是$A^{\mathsf{T}}A$的特征向量,考虑到逼近最优性，

$\mathbf{u}$，$\mathbf{v}$是SVD$: A = UDV^{\mathsf{T}}$中U,V的第一列。

例4(续)

```
> svd(A)
```

```
$U    [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 0.71 -0.53  0.00 -0.45  0.00
[2,] 0.56  0.53 -0.58  0.26  0.00
[3,] 0.28  0.27  0.58  0.13 -0.71
[4,] 0.28  0.27  0.58  0.13  0.71
[5,] 0.13 -0.53  0.00  0.84  0.00
```

```
$D
[1] 2.56 1.41 1 0.68  0
```

最大奇异值

```
$V
     [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 0.22  0.38 -0.58  0.38 -0.58
[2,] 0.28 -0.38  0.00 -0.67 -0.58
[3,] 0.50  0.00 -0.58 -0.29  0.58
[4,] 0.33 -0.76  0.00  0.57  0.00
[5,] 0.72  0.38  0.58  0.09  0.00
```

网页hub排序(U的第一列):

$$a > b > c = d > e$$

网页权威性排序 (V的第一列):

$$e > c > d > b > a$$

PageRank算法（google）

HITS算法的缺点是将一个网页的Authority重要性只与到访网页的hub重要性关联，而没有与到访网页的Authority重要性直接关联。谷歌的PageRank算法也是只使用链接情况而不使用网页内容对网页排序，但不区分auth,hub。

邻接矩阵*A*将所有网页之间的链接/关联性等同看待，但事实上某个网页*w*被一个对外有很多链接的网页(即hub)链接并不一定说明*w*重要，也即一个网页对外链接越多（hub性质越强),它对提升被链接的网页的重要性的作用反而越弱。考虑到这一点，Page and Brin (1998) 把每一个链出网页的重要性平均分配到它链接的网页中。

PageRank的基本思想是：
重要的网页应该链接重要的网页，而且被对外链接较少的重要网页链接。这样一个自循环的描述其实就是特征向量在变换下的不变性。

PageRank 算法

假设网页i的重要性为$x_i, \mathbf{x}=(x_1,\dots,x_n)^{\mathrm{T}}$, PageRank假设：

$$x_i=\sum_{k:k\to i}\frac{x_k}{d_k}, \qquad (*)$$

网页i的x_i等于与之链接的所有网页的重要性x_k的加权和, 权重为$1/d_k$,
$d_k=$网页k链出个数，即邻接矩阵A的第k行的行和。

设A为邻接矩阵, 令各个网站对外链接的个数为 $\mathbf{d}=(d_1,\dots,d_n)^{\mathrm{T}}$,
即$\mathbf{d}=A\mathbf{1}_n$ （A的行和), 令 $D=diag(\mathbf{d})$, 则假设（*）为

$$\mathbf{x}=A^{\mathrm{T}}D^{-1}\mathbf{x} \qquad (**)$$

$\mathbf{x}$是$A^{\mathrm{T}}D^{-1}$的特征根1对应的特征向量。

~~练习：证明$H=D^{-1}A$的特征根为实数;~~
~~证明1是H的特征根, 且是最大特征根。~~

A对称时该结论才成立。
这里的A不对称。

PageRank算法：

邻接矩阵为$A, D = diag(A\mathbf{1}_n), H = D^{-1}A,$

记x_i为网页i的重要性得分$(score)$, $PageRank$假设模型：$\mathbf{x} = H^{\top}\mathbf{x}$

迭代求解**x**：　$\mathbf{x}^{(k+1)} = H^{\top}\mathbf{x}^{(k)}, k = 0,1,2,....$

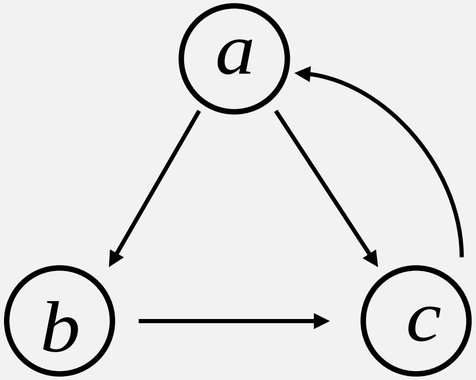
收敛到最大特征根的特征向量。

注1. PageRank对H做了修正：

$$Google\ matrix\ \ G = \alpha H + (1-\alpha)\mathbf{1}_n\mathbf{1}_n^{\top} / n$$

注2. 鉴于网页数目巨大，一般的求解特征向量的方法都不适用。Google使用的幂次迭代计算方法。

例5. 假设如下三个网页



链入

$$\text{链出}\quad A=\begin{array}{cccc} & a & b & c \\ a & 0 & 1 & 1 \\ b & 0 & 0 & 1 \\ c & 1 & 0 & 0 \end{array}$$

1. HITS算法

单位化的$\mathbf{v}=(0,\ 0.53,\ 0.85)$,　所以HITS权威性排序：$c>b>a$

2. PageRank算法

a,b,c的重要性得分，排序为

$$x_a=x_c=0.4>x_b=0.2,$$

即a,c同等重要,比HITS的 排序更合理。