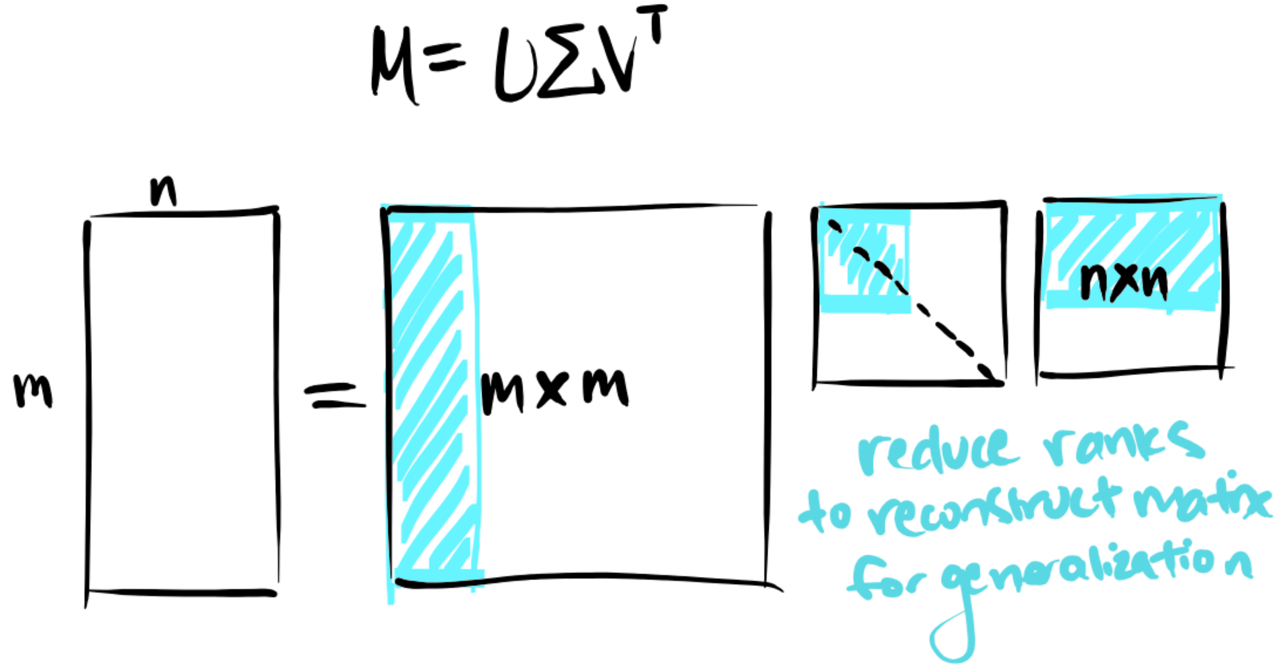
奇异值分解

Singular Value Decomposition



赵海涛

haitaozhao@ecust.edu.cn

大纲

- 奇异值分解定理
- 奇异值分解的性质
- 奇异值分解的应用：推荐系统

奇异值分解定义

矩阵奇异值分解是指，将一个非零的秩为r的$m \times n$实矩阵$\boldsymbol{A}$，$\boldsymbol{A} \in \mathcal{R}^{m \times n}$，表示为一下三个实矩阵乘积形式的运算，即进行矩阵的因子分解：

$$\boldsymbol{A} = \boldsymbol{P\Sigma Q}^T$$

其中，$\boldsymbol{P} \in \mathcal{R}^{m \times m}$，$\boldsymbol{Q} \in \mathcal{R}^{n \times n}$为正交矩阵，$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathcal{R}^{m \times n}$

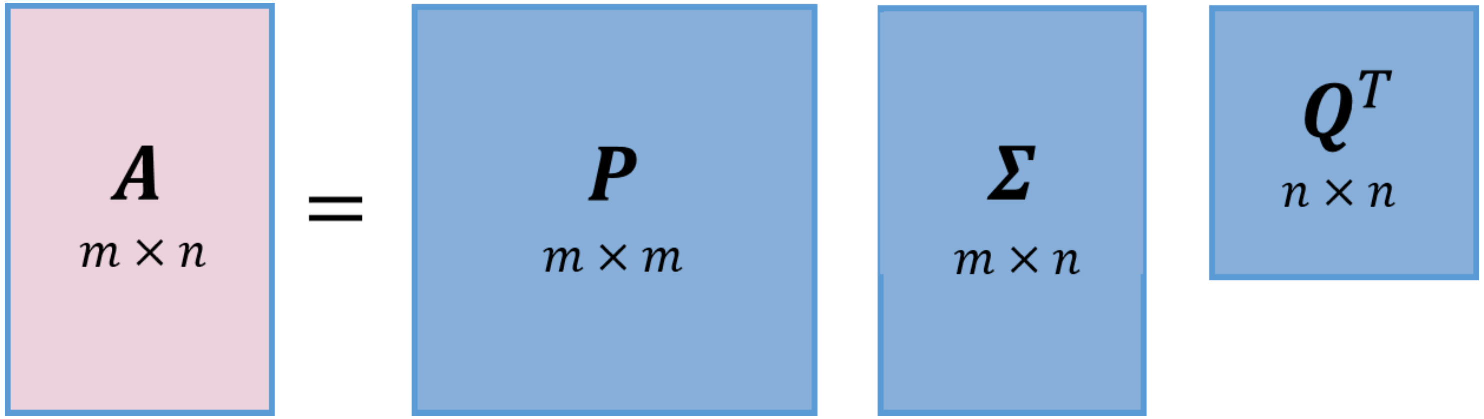
并且 $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r)$，$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

奇异值分解定义

$$\underset{m \times n}{\boldsymbol{A}} = \underset{m \times m}{\boldsymbol{P}} \; \underset{m \times n}{\boldsymbol{\Sigma}} \; \underset{n \times n}{\boldsymbol{Q}^T}$$

- 不失一般性，仅考虑$m \geq n$的情况

奇异值分解定义



- σ_i：矩阵$\boldsymbol{A}$的奇异值（singular value)
- $\boldsymbol{P}$的列向量：左奇异向量（left singular vector)
- $\boldsymbol{Q}$ 的列向量：右奇异向量（right singular vector)

奇异值分解定理

如果$\boldsymbol{A}$是一个非零的秩为r的$m\times n$实矩阵，$\boldsymbol{A}\in\mathcal{R}^{m\times n}$，则$\boldsymbol{A}$ 的奇异值分解存在：

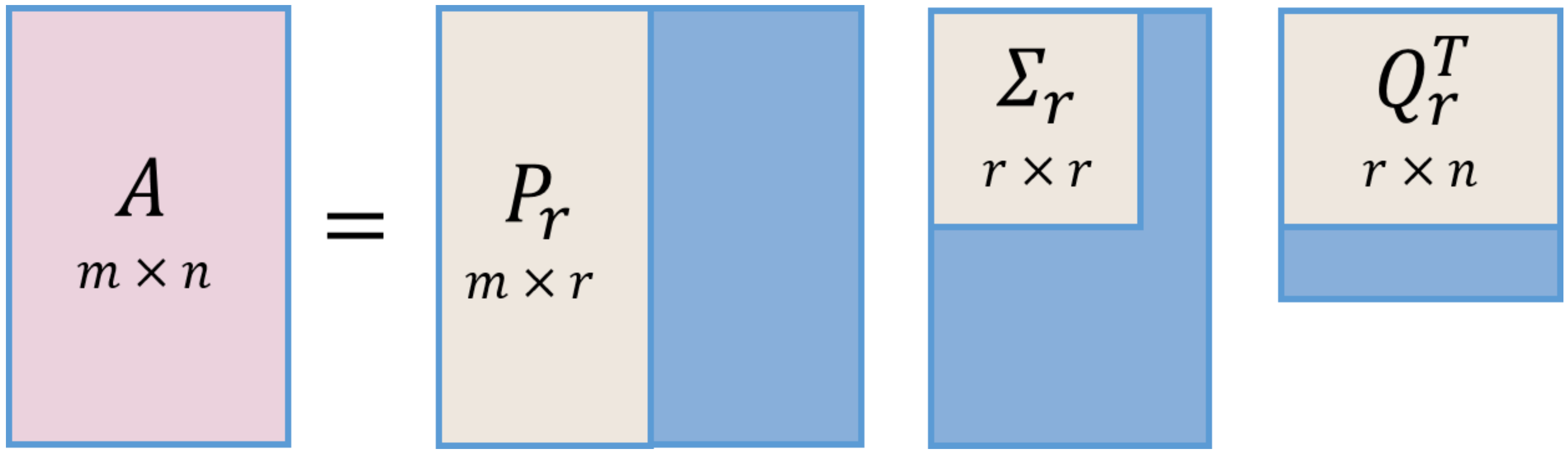
$$\boldsymbol{A}=\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{Q}^T$$

其中，$\boldsymbol{P}\in\mathcal{R}^{m\times m}$，$\boldsymbol{Q}\in\mathcal{R}^{n\times n}$为正交矩阵，$\boldsymbol{\Sigma}=\begin{bmatrix}\boldsymbol{\Sigma}_r & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\in\mathcal{R}^{m\times n}$

并且 $\boldsymbol{\Sigma}_r=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_r)$，$\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_r>0.$

（证明过程略）

奇异值分解紧凑形式

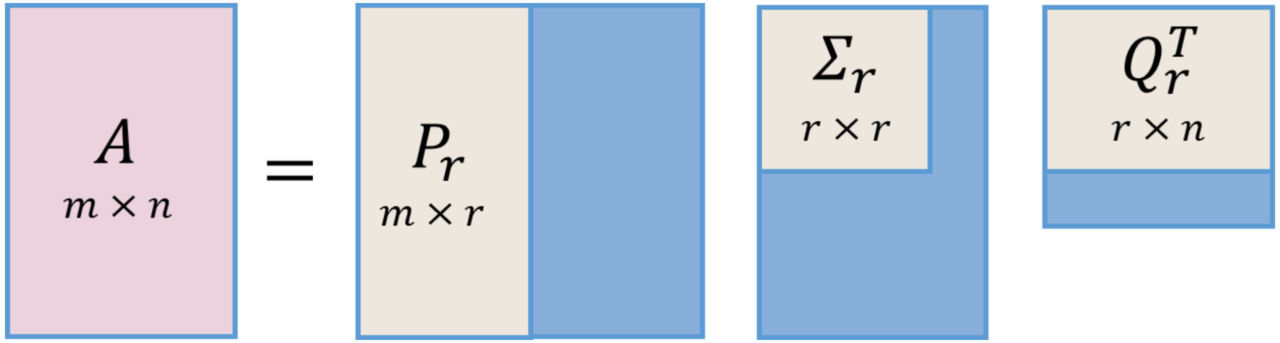


- 根据奇异值分解定理，$\boldsymbol{A}$的奇异值分解可以写成更紧凑的形式

$$\boldsymbol{A}=\boldsymbol{P}_r\boldsymbol{\Sigma}_r\boldsymbol{Q}_r{}^T$$

其中 $\boldsymbol{P}_r$ 和 $\boldsymbol{Q}_r$ 分别是$\boldsymbol{P}$ 和 $\boldsymbol{Q}$的前r列向量构成的矩阵

奇异值分解紧凑形式



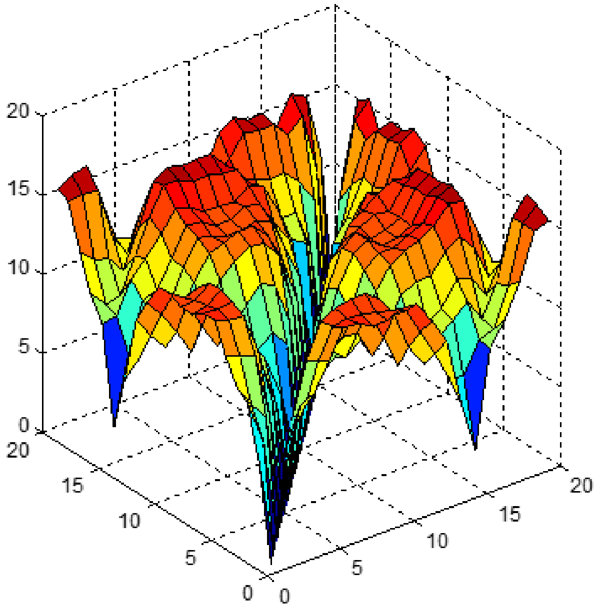
- $\{\sigma_1,\sigma_2,\cdots,\sigma_r\}$ 由矩阵$\boldsymbol{AA}^T$ (or $\boldsymbol{A}^T\boldsymbol{A}$)的特征值的非负平方根组成
- 奇异值向量不唯一，但也绝不是任意的。$\boldsymbol{P}_r$的列向量构成了$\boldsymbol{A}$的列空间的一组标准正交基， 而$\boldsymbol{Q}_r$ 的列向量构成$\boldsymbol{A}^T$的列空间的一组标准正交基
- 矩阵$\boldsymbol{AA}^T$ (or $\boldsymbol{A}^T\boldsymbol{A}$)的特征向量与$\boldsymbol{P}_r$(or $\boldsymbol{Q}_r$)
- $\boldsymbol{P}_r$与 $\boldsymbol{Q}_r$的关系与正交投影的关系
- 奇异值分解与解线性方程组

奇异值分解的性质

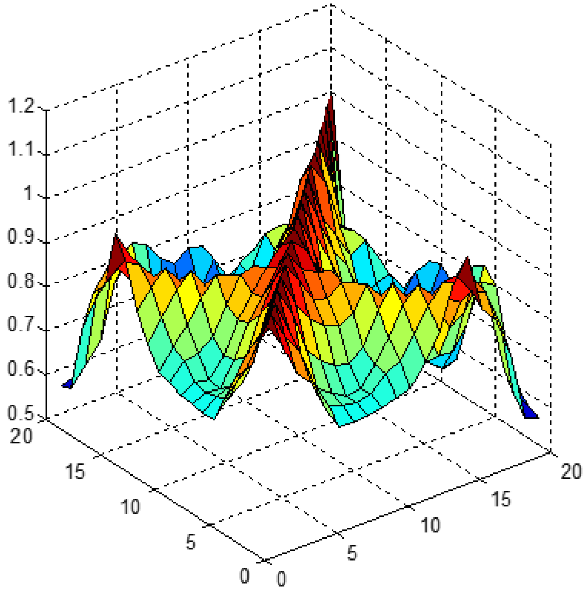
- We select one image for each object and the total number of images is 20. The resolution of each image is 128×128. We reshape the images to column vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$, $\boldsymbol{a}_i \in \mathcal{R}^{16384}$ and obtain the data matrix

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n] \in \mathcal{R}^{16384 \times 20}.$$

奇异值分解的性质



(a) $\|\boldsymbol{a}_i - \boldsymbol{a}_j\|$



(a) $\cos\left(\theta\left(\boldsymbol{a}_i, \boldsymbol{a}_j\right)\right) = \frac{\boldsymbol{a}_i^T \boldsymbol{a}_j}{\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|}$

奇异值分解的性质

- 令$\boldsymbol{Z}_r = \boldsymbol{\Sigma}_r \boldsymbol{Q}_r{}^T$。 易见$\boldsymbol{Z}_r \in \mathcal{R}^{r\times n}$。 令 $\boldsymbol{Z}_r = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n]$， $\boldsymbol{z}_i \in \mathcal{R}^r$.
- $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n\}$ 保持了$\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$的几何结构
  - ✓ 当从原有m维数据中提取r维特征，原始数据对之间的距离和角度（相关系数）可以被保留

$$\|\boldsymbol{z}_i - \boldsymbol{z}_j\| = \|\boldsymbol{a}_i - \boldsymbol{a}_j\|$$

$$\frac{\boldsymbol{z}_i^T \boldsymbol{z}_j}{\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|} = \frac{\boldsymbol{a}_i^T \boldsymbol{a}_j}{\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|}$$

奇异值分解的性质

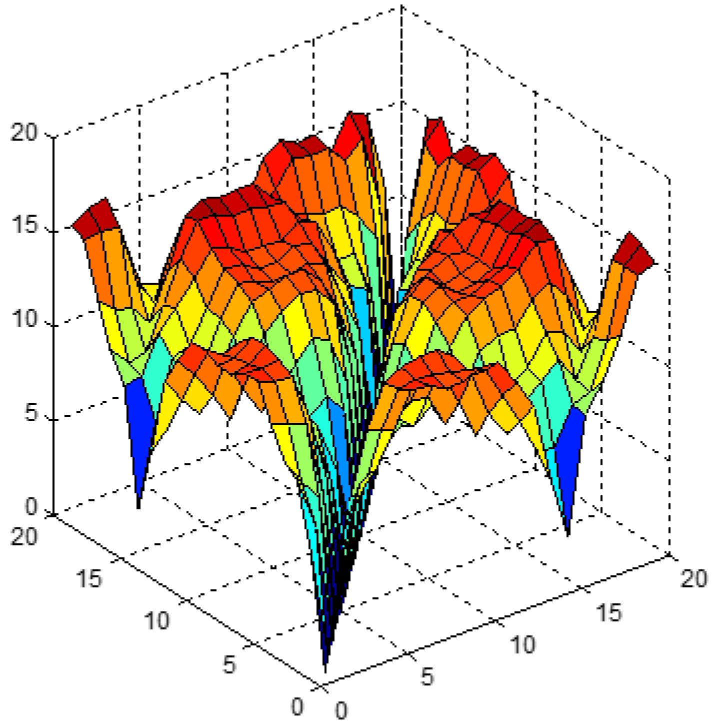
$$\left\|\boldsymbol{a}_i - \boldsymbol{a}_j\right\|^2 = \left\|\boldsymbol{P}_r(\boldsymbol{z}_i - \boldsymbol{z}_j)\right\|^2$$

$$= (\boldsymbol{z}_i - \boldsymbol{z}_j)^T \boldsymbol{P}_r^T \boldsymbol{P}_r (\boldsymbol{z}_i - \boldsymbol{z}_j) = \left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|^2$$

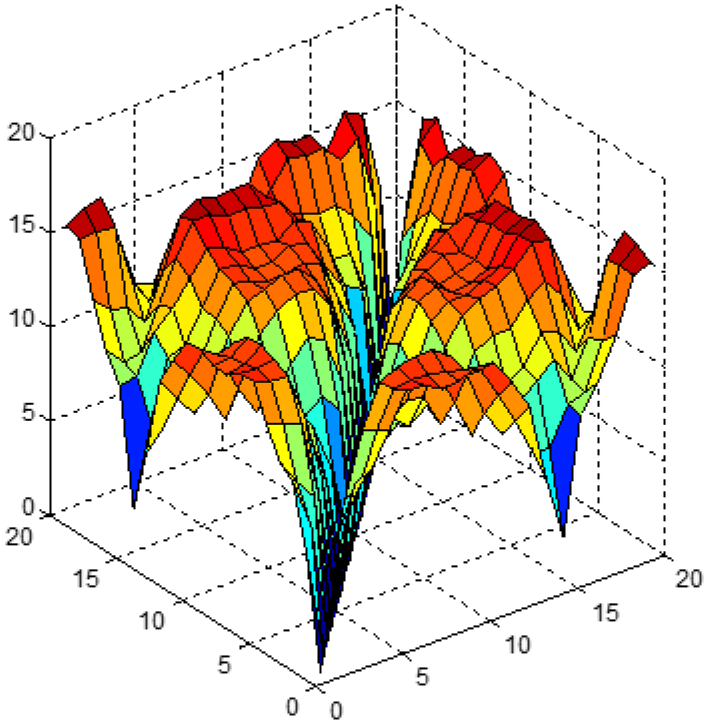
$$\frac{\boldsymbol{a}_i^T \boldsymbol{a}_j}{\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|} = \frac{\boldsymbol{z}_i^T \boldsymbol{P}_r^T \boldsymbol{P}_r \boldsymbol{z}_j}{\|\boldsymbol{P}_r \boldsymbol{z}_i\|\|\boldsymbol{P}_r \boldsymbol{z}_j\|}$$

$$= \frac{\boldsymbol{z}_i^T \boldsymbol{P}_r^T \boldsymbol{P}_r \boldsymbol{z}_j}{\sqrt{\boldsymbol{z}_i^T \boldsymbol{P}_r^T \boldsymbol{P}_r \boldsymbol{z}_i}\sqrt{\boldsymbol{z}_j^T \boldsymbol{P}_r^T \boldsymbol{P}_r \boldsymbol{z}_j}} = \frac{\boldsymbol{z}_i^T \boldsymbol{z}_j}{\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|}$$

奇异值分解的性质

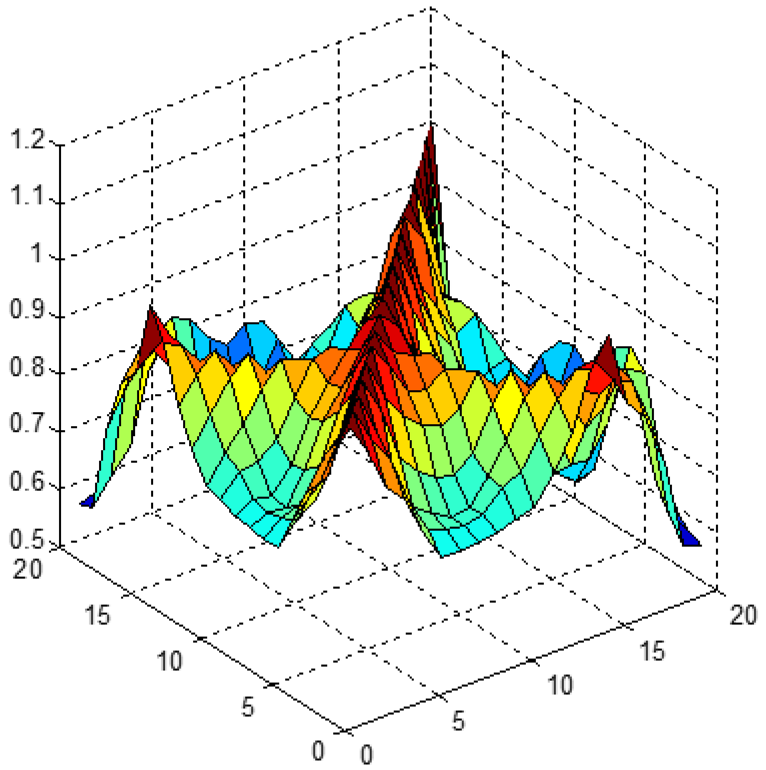


(a) $\|\boldsymbol{a}_i - \boldsymbol{a}_j\|$

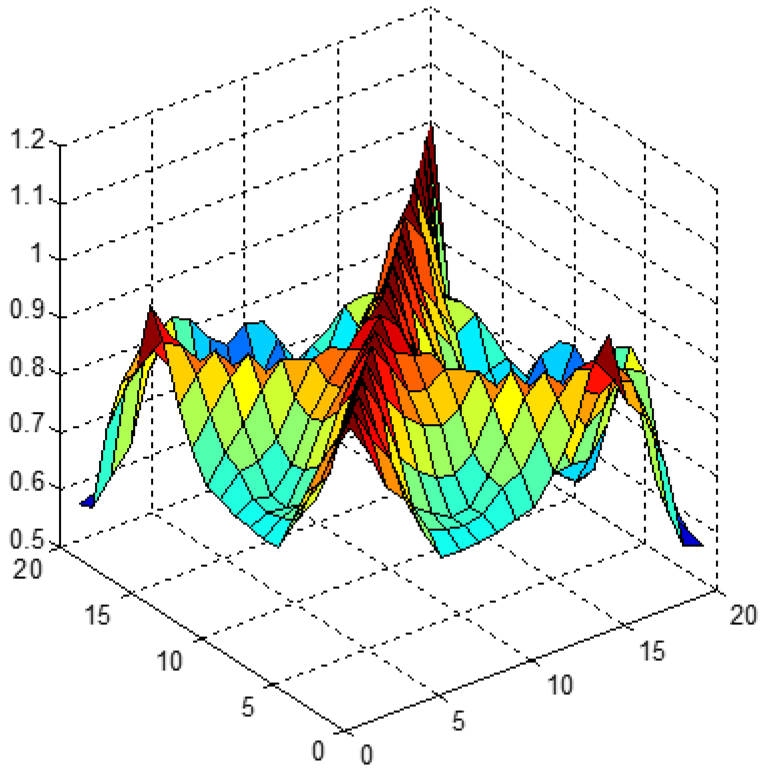


(b) $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|$

奇异值分解的性质



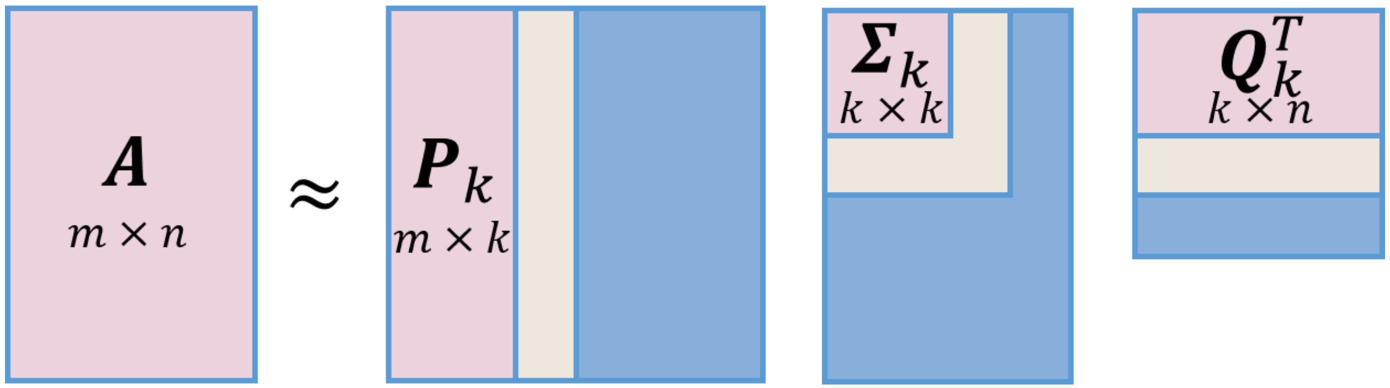
(a) $\cos\left(\theta\left(\boldsymbol{a}_i,\boldsymbol{a}_j\right)\right)=\dfrac{\boldsymbol{a}_i^T\boldsymbol{a}_j}{\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|}$



(b) $\cos\left(\theta\left(\boldsymbol{z}_i,\boldsymbol{z}_j\right)\right)=\dfrac{\boldsymbol{z}_i^T\boldsymbol{z}_j}{\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|}$

截断奇异值分解

truncated singular value decomposition
reduced-dimension representation (RDR)



- 奇异值分解中，只取最大的k个奇异值（$k < r$, r为矩阵的秩）对应的部分，就得到矩阵的截断奇异值分解。
- 实际应用中提到矩阵的奇异值分解时，通常指截断奇异值分解。
- 截断误差

$$\left\|\boldsymbol{A} - \boldsymbol{P}_k \boldsymbol{\Sigma}_k {\boldsymbol{Q}_k}^T\right\|_F = \left(\sum_{i=k+1}^{r} \sigma_i^2\right)^{\frac{1}{2}}$$

截断奇异值分解

card.jpg(2976×2976)

奇异值分解的计算

- 可通过计算矩阵$\boldsymbol{A}\boldsymbol{A}^T$ (or $\boldsymbol{A}^T\boldsymbol{A}$)的特征分解来求解，eig(A*A’) (or eig(A’*A))
- MATLAB中可以直接使用 [U,S,V] = svd(A)来计算
- Python中 U, D, VT = np.linalg.svd(A)

奇异值分解的例子

$$A = \begin{bmatrix} 2 & 3 & 2 & 0 & 0 \\ 3 & 2 & 3 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 0 & 1 & 0 & 4 & 5 \\ 0 & 2 & 0 & 5 & 4 \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix}$$

Given a rating matrix $\boldsymbol{A}$, in which each row in the matrix represents a reader's rating to different books. Suppose the books include three fantasy books: *The Lord of Rings*, *Harry Potter* and *A Game of Thrones*, and two machine learning books: *Feature Learning* and *Deep Learning*. Seven readers rated five books. The rating score was selected from one to five, and score zero means that the user has not rated the book. For example, the first reader rated the five books with [2 3 2 0 0]. He/She read three fantasy books and rated them with 2, 3, 2, respectively. He/She did not read or rate the machine learning books.

奇异值分解的例子

$$A = \begin{bmatrix} 2 & 3 & 2 & 0 & 0 \\ 3 & 2 & 3 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 0 & 1 & 0 & 4 & 5 \\ 0 & 2 & 0 & 5 & 4 \\ 0 & 0 & 0 & 3 & 3 \end{bmatrix}$$

- Perform SVD on $\boldsymbol{A}$ and get $\boldsymbol{\Sigma} = \mathrm{diag}(12.82, 9.98, 1.55, 0.88, 0, 0, 0)$. It means the rank of $\boldsymbol{A}$ is 4 and $\boldsymbol{\Sigma}_r = \mathrm{diag}(12.82, 9.98, 1.55, 0.88)$. We also can get

$$\boldsymbol{P}_r = \begin{bmatrix} 0.31 & -0.06 & 0.48 & 0.36 \\ 0.35 & -0.09 & -0.49 & -0.34 \\ 0.67 & -0.15 & -0.01 & 0.02 \\ 0.53 & -0.12 & -0.01 & 0.02 \\ 0.13 & 0.62 & -0.30 & 0.66 \\ 0.18 & 0.63 & 0.53 & -0.48 \\ 0.06 & 0.41 & -0.40 & -0.29 \end{bmatrix} \qquad \boldsymbol{Q}_r = \begin{bmatrix} 0.56 & -0.17 & -0.37 & -0.15 \\ 0.59 & 0.03 & 0.75 & 0.31 \\ 0.56 & 0.17 & -0.37 & -0.15 \\ 0.12 & 0.69 & 0.16 & -0.70 \\ 0.12 & 0.69 & -0.37 & -0.61 \end{bmatrix}$$

奇异值分解的例子

- Since the nonzero singular values 1.55 and 0.88 are much smaller than the first two singular values:

$$\frac{1.55 + 0.88}{12.82 + 9.98 + 1.55 + 0.88} = 0.096$$

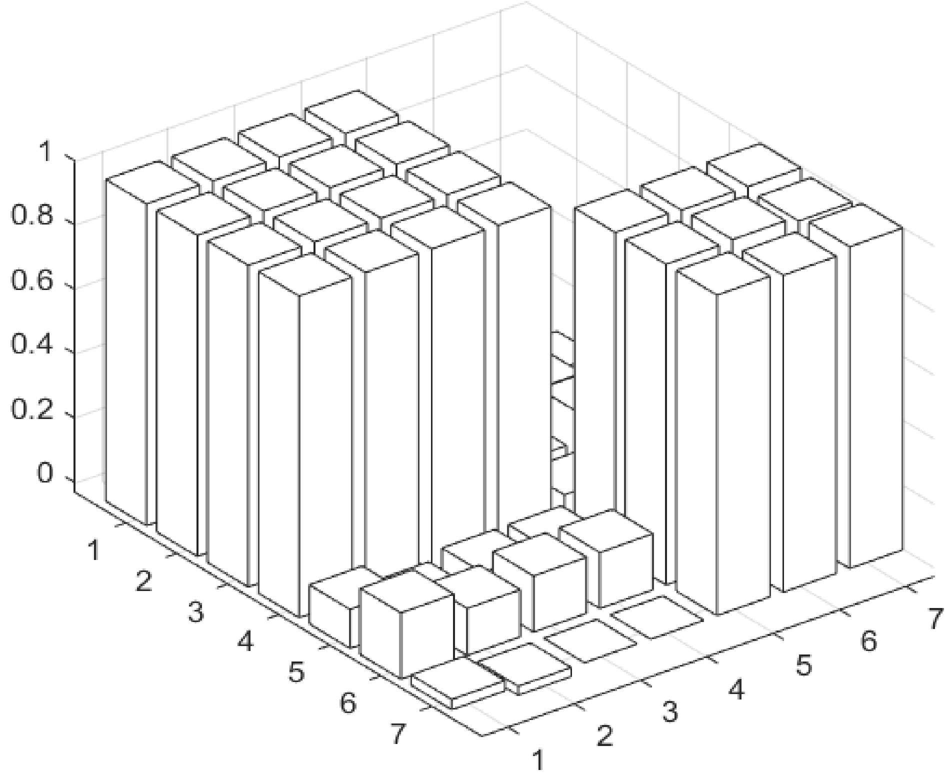
- we select $k = 2$ and obtain $\boldsymbol{\Sigma}_2 = \mathrm{diag}(12.82, 9.98)$
- Then the feature matrix $\boldsymbol{D}_2$ can be obtained as

$$\boldsymbol{D}_2 = \boldsymbol{P}_2\boldsymbol{\Sigma}_2 = \begin{bmatrix} 4.00 & -0.58 \\ 4.52 & -0.94 \\ 8.53 & -1.52 \\ 6.82 & -1.21 \\ 1.69 & 6.21 \\ 2.29 & 6.24 \\ 0.73 & 4.12 \end{bmatrix}$$

奇异值分解的例子

- Compute the correlation coefficient as the similarity of feature $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, i.e

$$s_{ij} = \frac{\boldsymbol{d}_i^T \boldsymbol{d}_j}{\|\boldsymbol{d}_i\| \|\boldsymbol{d}_j\|}.$$



奇异值分解的例子

- two new readers whose rating are as below:

$$\boldsymbol{r}_1 = [0 \quad 4 \quad 0 \quad 0 \quad 0]$$

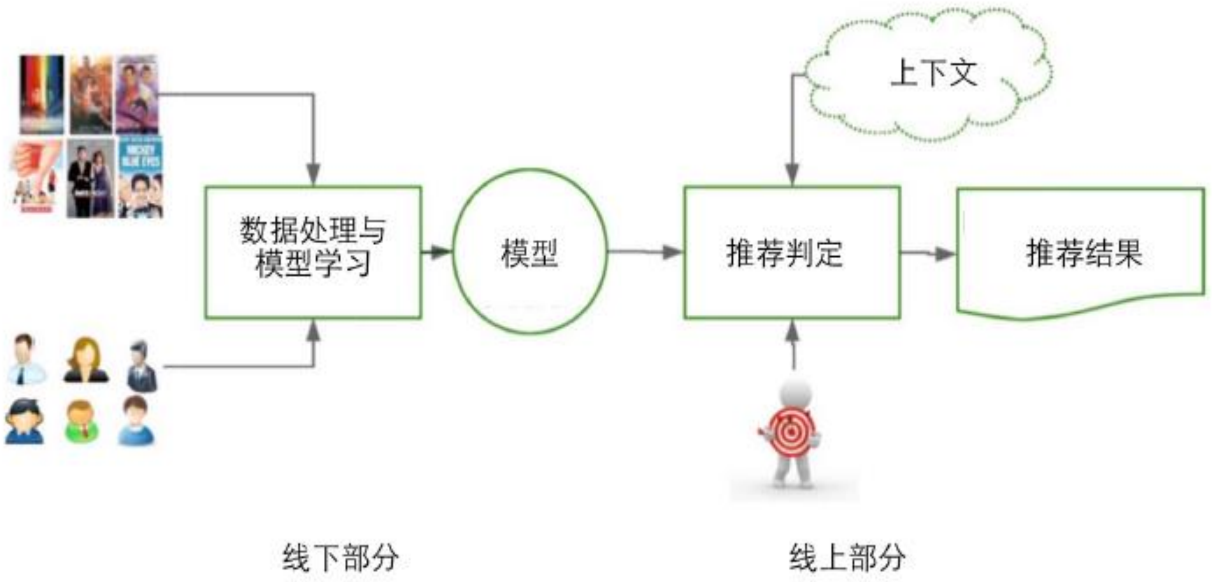
$$\boldsymbol{r}_2 = [5 \quad 0 \quad 1 \quad 0 \quad 0]$$

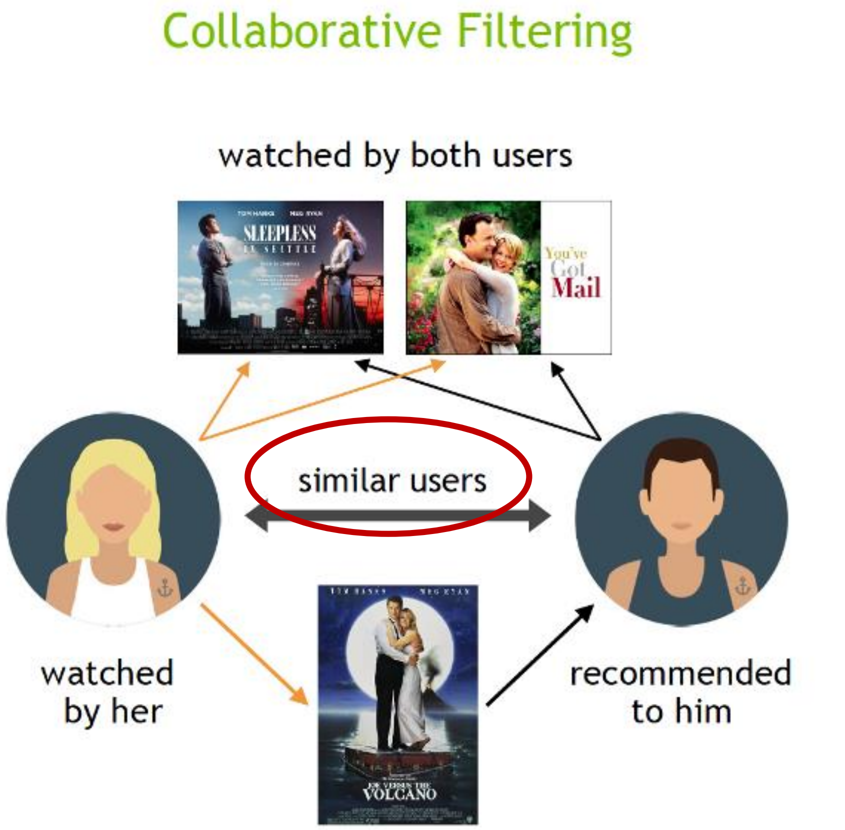
- It is easy to compute the similarity as $\frac{\boldsymbol{r}_1^T \boldsymbol{r}_2}{\|\boldsymbol{r}_1\|\|\boldsymbol{r}_2\|} = 0$, which means these two readers have no common reading interest. However, both these two readers rated fantasy books with high scores and have no rating score on machine learning books. The similarity of the features after rank-2 approximation is

$$\frac{\boldsymbol{Q}_2{}^T \boldsymbol{r}_1^T \boldsymbol{r}_2 \boldsymbol{Q}_2}{\|\boldsymbol{Q}_2{}^T \boldsymbol{r}_1\|\|\boldsymbol{Q}_2{}^T \boldsymbol{r}_2\|} = 0.95$$

奇异值分解的例子

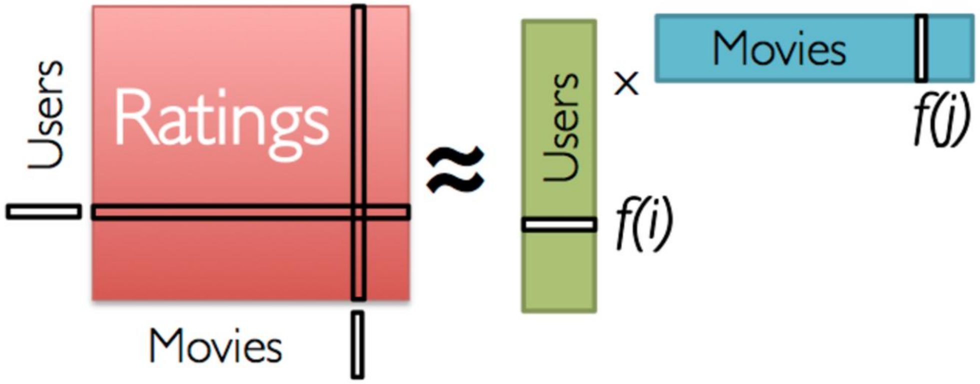
- 电影推荐系统 (基于用户与基于商品)





奇异值分解的例子

- 电影推荐系统 (基于用户与基于商品)
- SVD/SVD++



User No.	Movie No.	The rating	Timestamp
1	1	5	874965758
1	2	3	876893171
1	3	4	878542960
…	…	…	…
2	1	4	888550871
2	10	2	888551853
…	…	…	…

SVD++

- 假设已知的评分为r_{ui}，那么真实值与预测值的误差为：

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

继而有总的误差平方和：

$$\mathrm{SSE} = \frac{1}{2}\sum_{u,i} e_{ui}^2 = \frac{1}{2}\sum_{u,i}\left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ki}\right)^2$$

所以，只要使得上式降到最小，P和Q就和很好地拟合R了。

SVD++（续）

- 梯度下降法求解（只推导p_{uk}）

$$\frac{\partial}{\partial p_{uk}}(\mathrm{SSE}) = \frac{\partial}{\partial p_{uk}}\left(\frac{1}{2}e_{ui}^2\right) = e_{ui}\frac{\partial}{\partial p_{uk}}e_{ui}$$

$$= e_{ui}\frac{\partial}{\partial p_{uk}}\left(r_{ui} - \sum_{K=1}^{K} p_{uk}q_{ki}\right) = -e_{ui}q_{ki}$$

$$p_{uk} := p_{uk} - \eta(-e_{ui}q_{ki}) = p_{uk} + \eta e_{ui}q_{ki}$$

- 对于Q同样有：

$$q_{ki} := q_{ki} - \eta(-e_{ui}p_{uk}) = q_{ki} + \eta e_{ui}p_{uk}$$

SVD++（续）

- 为了防止数据过拟合，可在目标函数中加入正则化参数，此时则有：

$$SSE=\frac{1}{2}\sum_{u,i}e_{ui}^2+\frac{1}{2}\lambda\sum_u|p_u|^2+\frac{1}{2}\lambda\sum_i|q_i|^2$$

- 通过求梯度，则可将递推公式更新如下：

$$p_{uk}:=p_{uk}+\eta(e_{ui}q_{ki}-\lambda p_{uk})$$

$$q_{ki}:=q_{ki}+\eta(e_{ui}p_{uk}-\lambda q_{ik})$$

电影推荐实验

- MovieLens 是历史最悠久的推荐系统。它由美国 Minnesota 大学计算机科学与工程学院的 GroupLens 项目组创办，是一个非商业性质的、以研究为目的的实验性站点。MovieLens 主要使用 Collaborative Filtering 和 Association Rules 相结合的技术，向用户推荐他们感兴趣的电影。
- 本次实验主要使用的数据文件是u1.base（训练集）和u1.test（测试集）。其中u1.base文件内容为4列数据，分别代表用户编号、电影编号、某用户对该电影的评分和时间戳，u1.test中的数据含义类似。需要说明的是，u1.base仅包含一些用户对部分电影的评分数据，而u1.test的评分数据则是填补了u1.base文件所缺的部分用户对电影的实际评分。因此可以通过某些算法预测出用户对未看过的电影评分，进而可根据预测评分高低来针对用户推荐电影。本次实验主要做的是前半部分（即预测评分），而后将预测的评分与u1.test中的评分对比，且通过计算两者的RMSE（均方根误差）来评价本次预测的好坏。

电影推荐实验

经过第二次课程的模式识别学习，在赵海涛老师的辛勤指导下，我对模式识别的 SVD 算法有了一定的了解，并通过本次针对 MovieLens 数据集的评分预测实验来巩固所学内容。

经过一段时间的程序编写及调试，并尝试了各种主流的推荐系统算法，最终将 RMSE 降至 0.91616 左右（程序测试在此范围会有小部分波动），跑完一次程序越 44 秒左右（视参数而定，基本在 1 分钟内，且能输出预测评分的 CSV 文档）。下面将详细说明我的解决过程。

电影推荐实验

四、作业总结

由于这是第一次接触稍大的数据处理，所以在算法实现上走了不少弯路：一开始我就建立了一个 943*1682 的较大稀疏矩阵，填入数据，再进行数据处理；在预测评分上，此前也有过将所有未评分的项目全部预测，因此大大增加计算量以及所耗费时间，某次程序最久跑了 15 个小时！甚至为了计算程序写了多线程。但是仍然不满意。

这次实验我尝试过采用 user-user，item-item，3 个相似度计算方法切换（包括欧氏距离法、余弦相似度法，Pearson 相似度法），预测评分上采用加权平均，或是相似度乘已知平方除以总相似度等方法，在最终预测成果上加权平均，老师所给程序的 Python 复现修改等等。但由于本人能力有限，并没有调试出更好的方案。所以最终结合结果和计算时间考虑，以此目前方法较优。

谢谢各位同学！