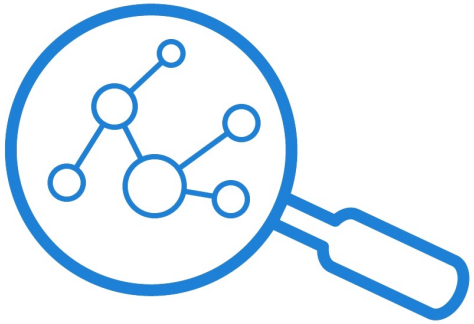
数据科学与大数据技术的数学基础

第十讲



计算机学院
余皓然

2024/5/23

课程内容

Part1 随机化方法
一致性哈希 布隆过滤器 CM Sketch方法 最小哈希 欧氏距离下的相似搜索 Jaccard相似度下的相似搜索
Part2 谱分析方法
主成分分析 奇异值分解 谱图论
Part3 最优化方法
压缩感知

奇异值分解

低秩矩阵近似

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \leq k.$$

$\|\cdot\|_F$是F-范数（Frobenius norm）：对任意$m \times n$矩阵$\mathbf{A}$，有$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{A}_{ij}^2}$

* 也有基于其余范数定义的低秩矩阵近似

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \le k.$$

$\|\cdot\|_F$是F-范数（Frobenius norm）：对任意$m\times n$矩阵$\mathbf{A}$，有$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$

为什么需要低秩近似?

* 也有基于其余范数定义的低秩矩阵近似

低秩矩阵近似

低秩近似的应用之一：数据压缩

- 对于原$m \times n$矩阵$\mathbf{A}$，直接存储则需记录mn个数字
- 若找到秩为k（$k < \mathrm{rank}(\mathbf{A})$）的近似矩阵$\mathbf{B}$，最少需要记录　　个数字

低秩矩阵近似

低秩近似的应用之一：数据压缩

- 对于原$m\times n$矩阵$\mathbf{A}$，直接存储则需记录mn个数字
- 若找到秩为k（$k<\mathrm{rank}(\mathbf{A})$）的近似矩阵$\mathbf{B}$，最少需要记录　个数字

例：

$$\mathbf{B}=\begin{bmatrix}1&2&-1&2\\2&4&1&1\\-1&-2&-2&1\end{bmatrix}\xrightarrow{\text{初等行变换}}\begin{bmatrix}1&2&0&1\\0&0&1&-1\\0&0&0&0\end{bmatrix}$$

可得$\mathrm{rank}(\mathbf{B})=2$且任何行向量可以写成$(1,2,0,1),(0,0,1,-1)$的线性组合

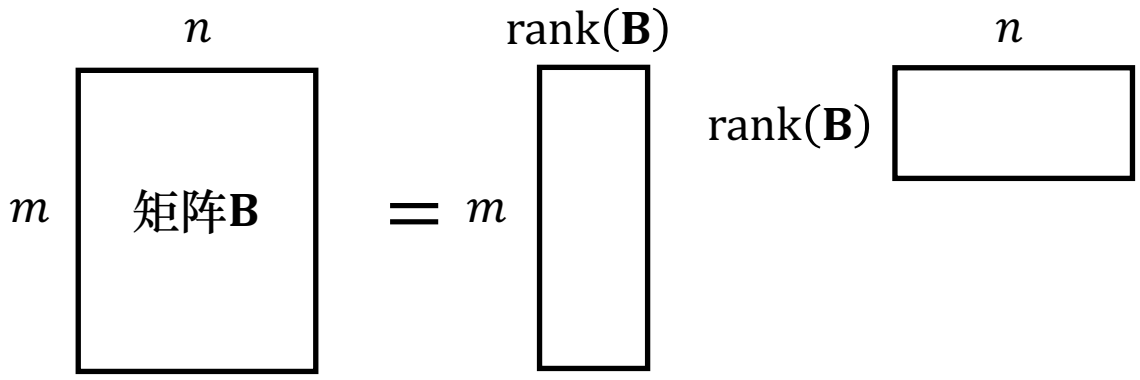
$$\mathbf{B}=\begin{bmatrix}1&2&-1&2\\2&4&1&1\\-1&-2&-2&1\end{bmatrix}=\begin{bmatrix}1&-1\\2&1\\-1&-2\end{bmatrix}\begin{bmatrix}1&2&0&1\\0&0&1&-1\end{bmatrix}$$

3×4　　$3\times\mathrm{rank}(\mathbf{B})$　　$\mathrm{rank}(\mathbf{B})\times 4$

低秩矩阵近似

低秩近似的应用之一：数据压缩

- 对于原$m\times n$矩阵**A**，直接存储则需记录mn个数字
- 若找到秩为k（$k<\mathrm{rank}(\mathbf{A})$）的近似矩阵**B**，最少需要记录　个数字



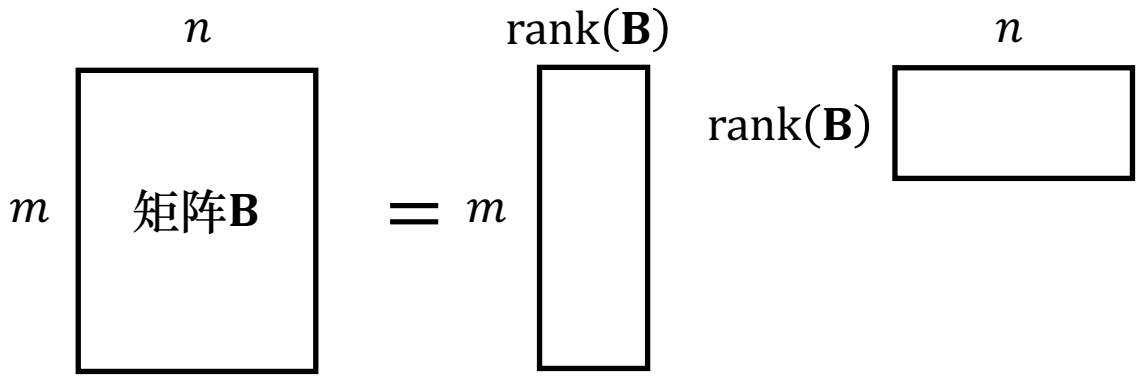
任意$m\times n$矩阵**B**可写为$m\times\mathrm{rank}(\mathbf{B})$矩阵与$\mathrm{rank}(\mathbf{B})\times n$矩阵相乘

（不能写为$m\times(\mathrm{rank}(\mathbf{B})-1)$矩阵与$(\mathrm{rank}(\mathbf{B})-1)\times n$矩阵相乘）

低秩矩阵近似

低秩近似的应用之一：数据压缩

- 对于原$m\times n$矩阵**A**，直接存储则需记录mn个数字　　假设$k \ll m, n$
- 若找到秩为k（$k < \mathrm{rank}(\mathbf{A})$）的近似矩阵**B**，最少需要记录$(m+n)k$个数字



任意$m\times n$矩阵**B**可写为$m\times\mathrm{rank}(\mathbf{B})$矩阵与$\mathrm{rank}(\mathbf{B})\times n$矩阵相乘

（不能写为$m\times(\mathrm{rank}(\mathbf{B})-1)$矩阵与$(\mathrm{rank}(\mathbf{B})-1)\times n$矩阵相乘）

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \le k.$$

$\|\cdot\|_F$是F-范数（Frobenius norm）：对任意$m\times n$矩阵$\mathbf{A}$，有$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}$

如何解决低秩近似问题?

低秩矩阵近似

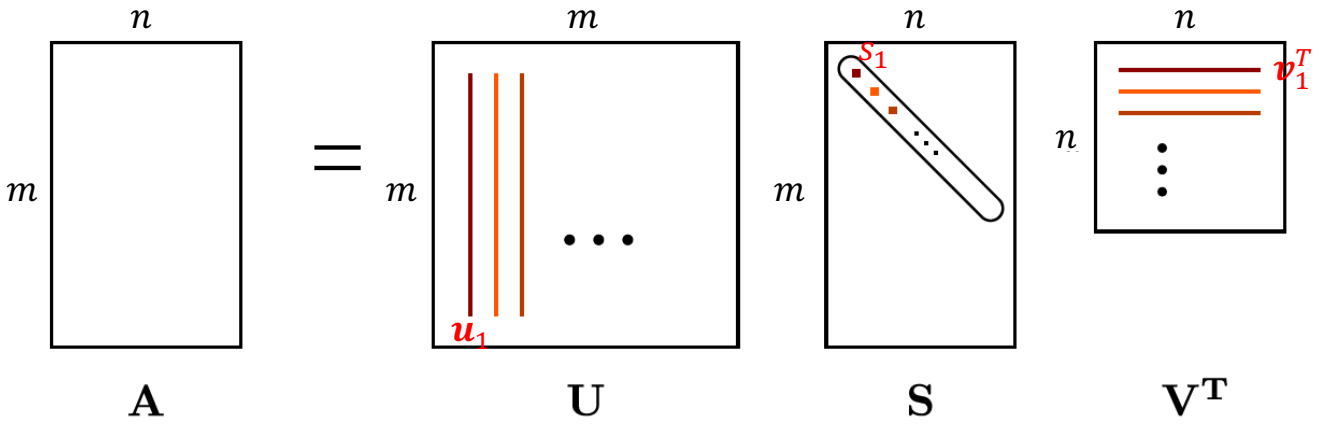
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t.} \quad \text{rank}(\mathbf{B}) \leq k.$$

回顾奇异值分解的等价形式：

$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ 等价于 $\mathbf{A} = \sum_{i=1}^{\min\{m,n\}} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$ （不妨设$s_1 \geq s_2 \geq \cdots$）



低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

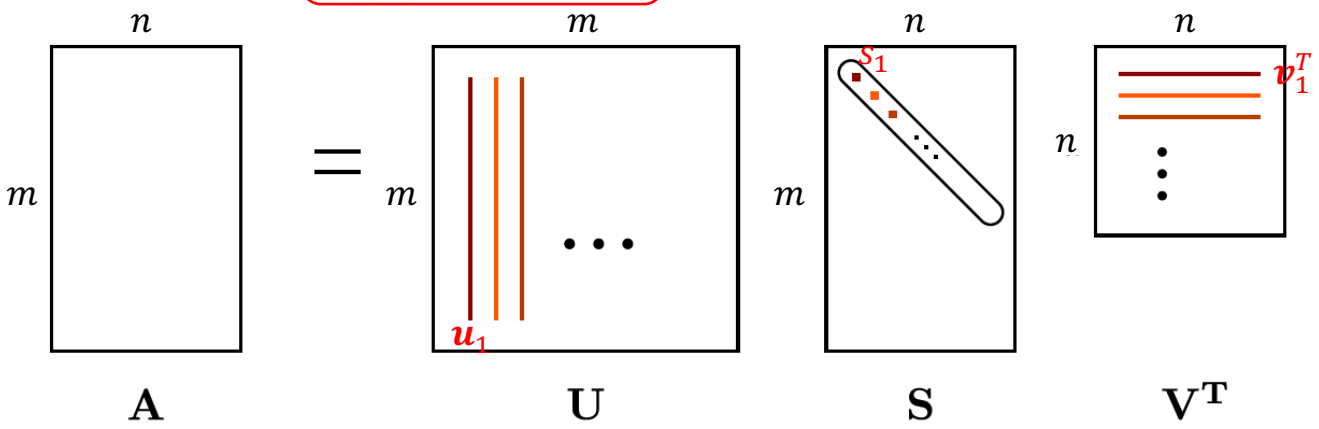
$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \le k.$$

回顾奇异值分解的等价形式：

$\mathbf{A}=\mathbf{USV}^T$ 等价于 $\mathbf{A}=\sum_{i=1}^{\min\{m,n\}} s_i\cdot\mathbf{u}_i\mathbf{v}_i^\top$

在$\min\{m,n\}$项中保留前k项，舍弃其余项

（不妨设$s_1\ge s_2\ge\cdots$）



低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

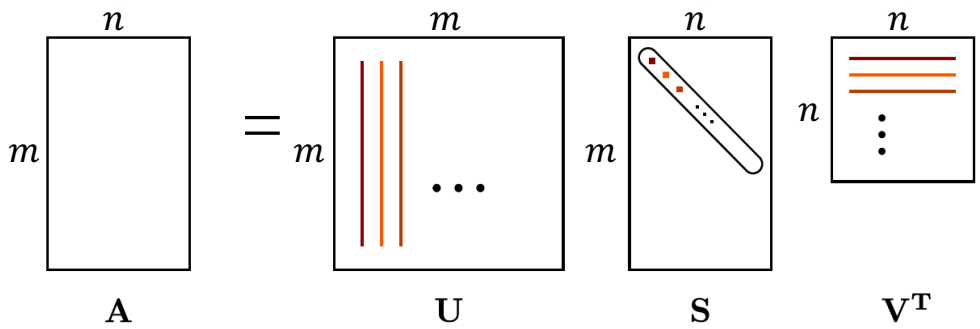
给定$m \times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t.} \quad \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么$\mathbf{B}$满足$\text{rank}(\mathbf{B}) \leq k$约束？

低秩矩阵近似



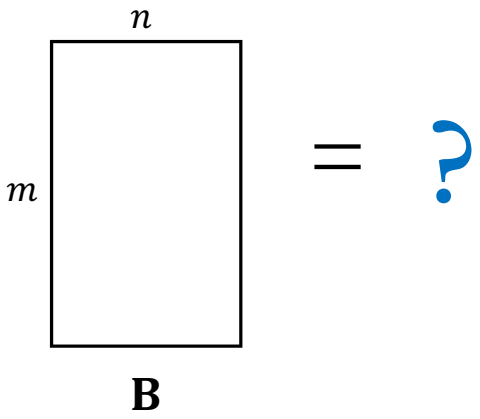
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么$\mathbf{B}$满足$\operatorname{rank}(\mathbf{B}) \leq k$约束?



低秩矩阵近似

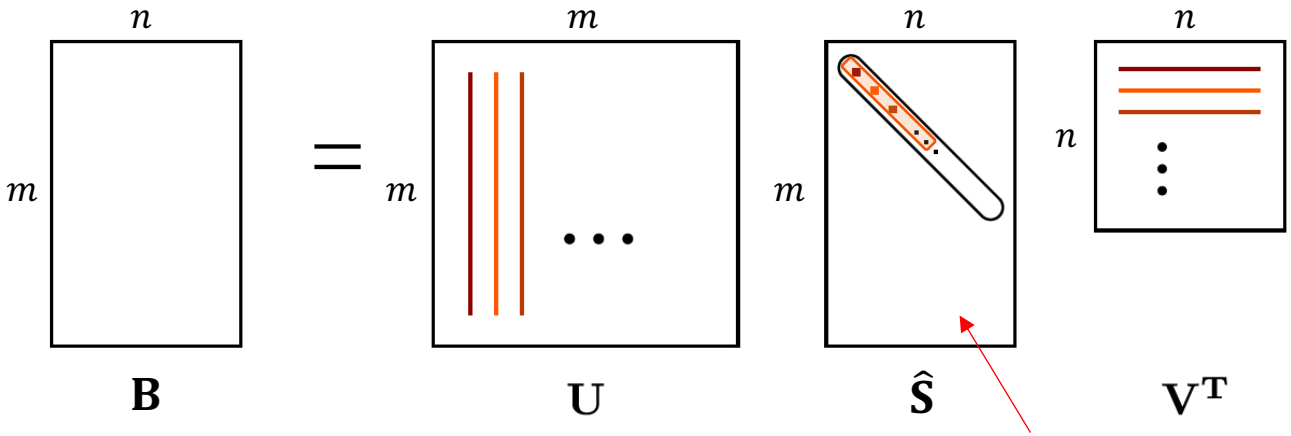
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵**A**及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵**B**为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么**B**满足$\text{rank}(\mathbf{B}) \leq k$约束?



保留对角线上前k个元素，其余（若为正）置0

（**U**和**V**不用变动）

低秩矩阵近似

例 $\mathbf{A}=\begin{bmatrix}\frac{1}{\sqrt{10}} & -\frac{3}{\sqrt{10}}\\ \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}}\end{bmatrix}\begin{bmatrix}\sqrt{45} & 0\\ 0 & \sqrt{5}\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}$

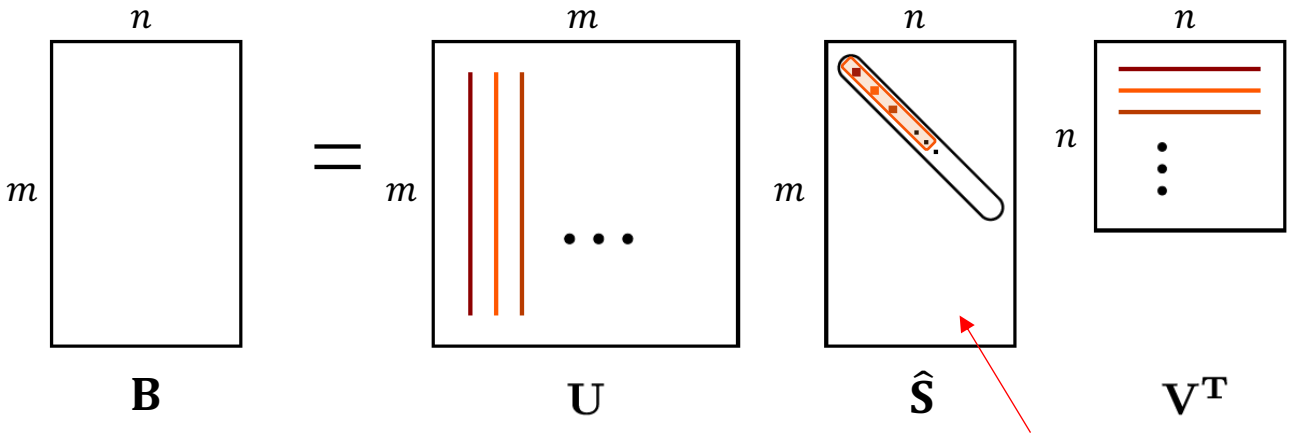
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

为什么$\mathbf{B}$满足$\operatorname{rank}(\mathbf{B})\le k$约束?



保留对角线上前k个元素，其余（若为正）置0

（U和V不用变动）

低秩矩阵近似

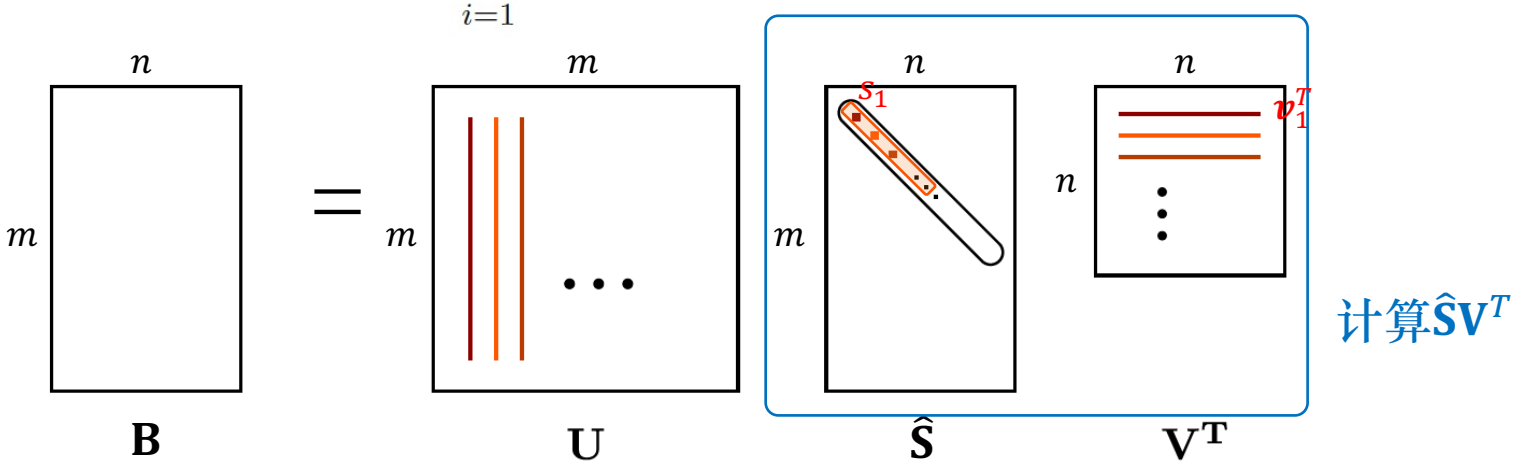
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵**A**及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵**B**为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么**B**满足$\text{rank}(\mathbf{B}) \leq k$约束?



计算$\hat{\mathbf{S}}\mathbf{V}^T$

低秩矩阵近似

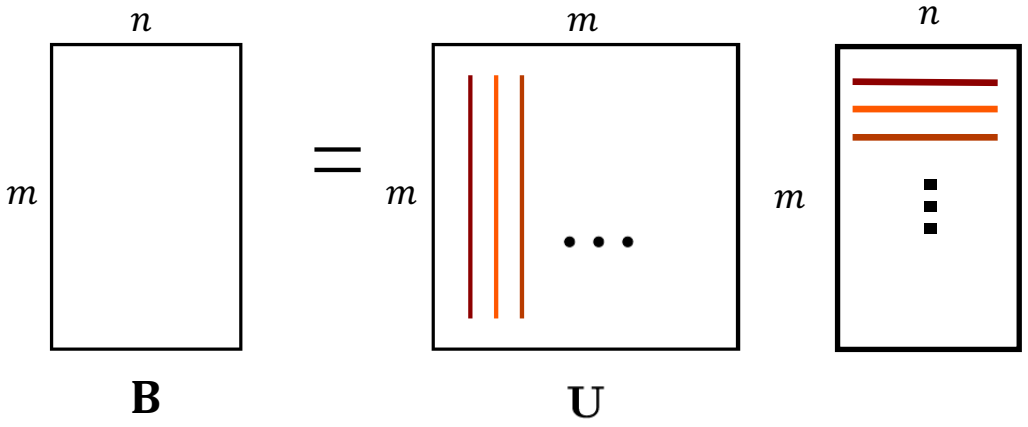
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵**A**及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵**B**为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么**B**满足$\text{rank}(\mathbf{B}) \leq k$约束?



前k行为$s_1 \boldsymbol{v}_1^T, \ldots, s_k \boldsymbol{v}_k^T$
剩余$m - k$行为零向量

低秩矩阵近似

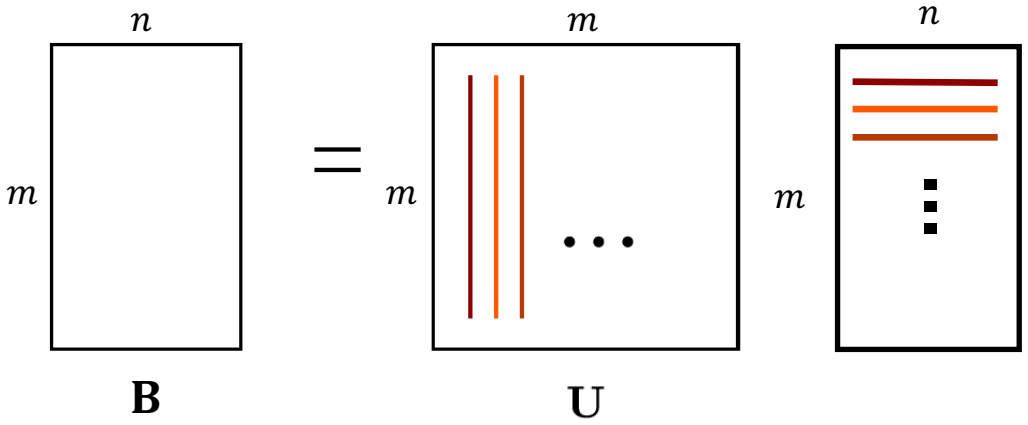
低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵**A**及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵**B**为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

为什么**B**满足$\text{rank}(\mathbf{B}) \leq k$约束?



前k行为$s_1 \boldsymbol{v}_1^T, \ldots, s_k \boldsymbol{v}_k^T$
剩余$m - k$行为零向量

矩阵**B**的所有行向量可表示为k个（可能线性相关的）行向量的线性组合，说明$\text{rank}(\mathbf{B}) \leq k$

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

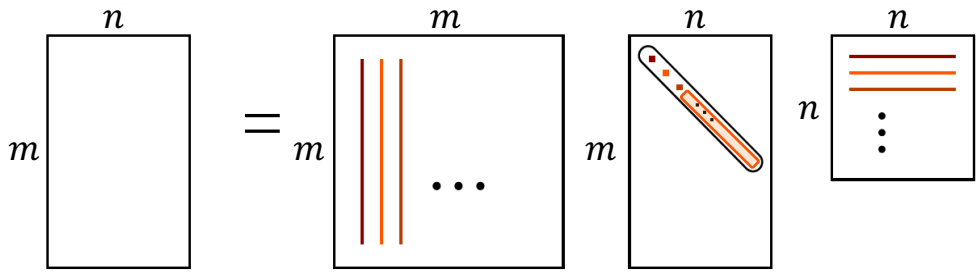
给定$m \times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t.} \quad \text{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^\top$

最优目标函数值?

低秩矩阵近似



低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

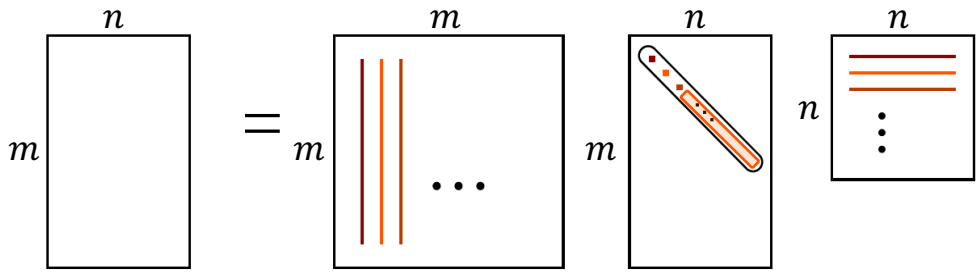
若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

最优目标函数值?

$$\|\mathbf{A}-\mathbf{B}\|_F=\left\|\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T-\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F=\left\|\sum_{i=k+1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F$$

一个新$m\times n$矩阵

低秩矩阵近似



低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

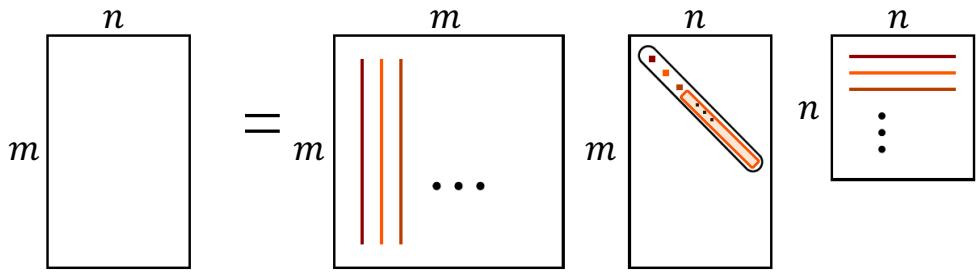
最优目标函数值?

$$\|\mathbf{A}-\mathbf{B}\|_F=\left\|\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T-\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F=\left\|\sum_{i=k+1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F$$

一个新$m\times n$矩阵

F-范数的性质：$\|\mathbf{A}\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{A}_{ij}^2}=\sqrt{\sum_{i=1}^{\min\{m,n\}}s_i^2}$

低秩矩阵近似



低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

最优目标函数值?

$$\|\mathbf{A}-\mathbf{B}\|_F=\left\|\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T-\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F=\left\|\sum_{i=k+1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F$$

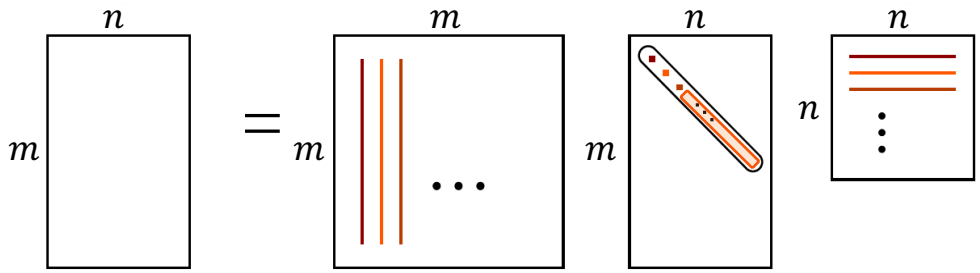
一个新$m\times n$矩阵

F-范数的性质：$\|\mathbf{A}\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{A}_{ij}^2}=\sqrt{\sum_{i=1}^{\min\{m,n\}}s_i^2}$

证明：先证明$\mathbf{tr}(\mathbf{A}^T\mathbf{A})=\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{A}_{ij}^2$，则有$\|\mathbf{A}\|_F=\sqrt{\mathbf{tr}(\mathbf{A}^T\mathbf{A})}$

再利用方阵的迹等于特征值之和（见第八讲对“近似误差”的分析）

低秩矩阵近似



低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

最优目标函数值?

$$\|\mathbf{A}-\mathbf{B}\|_F=\left\|\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T-\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F=\left\|\sum_{i=k+1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T\right\|_F=\sqrt{\sum_{i=k+1}^{\min\{m,n\}} s_i^2}$$

F-范数的性质：$\|\mathbf{A}\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{A}_{ij}^2}=\sqrt{\sum_{i=1}^{\min\{m,n\}} s_i^2}$

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

$$\|\mathbf{A}-\mathbf{B}\|_F=\sqrt{\sum_{i=k+1}^{\min\{m,n\}} s_i^2}$$

若$k=1$

例

$$\mathbf{A}=\begin{bmatrix}3 & 0\\ 4 & 5\end{bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{10}} & -\frac{3}{\sqrt{10}}\\ \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}}\end{bmatrix}\begin{bmatrix}\sqrt{45} & 0\\ 0 & \sqrt{5}\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix}\frac{3}{2} & \frac{3}{2}\\ \frac{9}{2} & \frac{9}{2}\end{bmatrix}$$

$$\|\mathbf{A}-\mathbf{B}\|_F=\sqrt{\left(\frac{3}{2}\right)^2+\left(\frac{3}{2}\right)^2+\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2}=\sqrt{5}$$

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

若$k<\min\{m,n\}$，取矩阵$\mathbf{B}$为 $\sum_{i=1}^{k} s_i\cdot\mathbf{u}_i\mathbf{v}_i^{\top}$

证明$\mathbf{B}$最优性的一种方法：

第一步：证明引理“若$\mathbf{Q}$为正交矩阵，有$\|\mathbf{QA}\|_F=\|\mathbf{A}\|_F$”（类比$\|\mathbf{Q}\boldsymbol{x}\|_2=\|\boldsymbol{x}\|_2$）

证明方法为利用$\|\mathbf{A}\|_F=\sqrt{\mathbf{tr}(\mathbf{A}^T\mathbf{A})}=\sqrt{\mathbf{tr}(\mathbf{A}^T\mathbf{Q}^T\mathbf{QA})}=\|\mathbf{QA}\|_F$

第二步：利用引理改写目标函数：$\|\mathbf{A}-\mathbf{B}\|_F=\|\mathbf{USV}^T-\mathbf{B}\|_F=\|\mathbf{S}-\mathbf{U}^T\mathbf{BV}\|_F$

第三步：说明因为$\mathbf{S}$是对角矩阵，对最优解$\mathbf{B}$，$\mathbf{U}^T\mathbf{BV}$必为对角矩阵

（反证，若$\mathbf{U}^T\mathbf{BV}$非对角阵，可找到同秩对角阵更小化$\mathbf{S}-\mathbf{U}^T\mathbf{BV}$元素平方和）

第四步：$\mathbf{B}$可取为$\mathbf{UDV}^T$，则目标函数为$\|\mathbf{S}-\mathbf{D}\|_F$。易证$k$秩约束下的最优$\mathbf{D}$

证明取自ccjou <线代启示录： SVD于矩阵近似的应用>

低秩矩阵近似

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m \times n$矩阵**A**及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \mathbf{B}\|_F$$
$$\text{s.t. } \operatorname{rank}(\mathbf{B}) \leq k.$$

若$k < \min\{m, n\}$，取矩阵**B**为 $\sum_{i=1}^{k} s_i \cdot \mathbf{u}_i \mathbf{v}_i^{\top}$

利用SVD对矩阵**A**进行低秩近似的步骤：

（1）对矩阵**A**做奇异值分解；

（2）计算矩阵$\mathbf{B} = \sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T$。

如何选择k？ 如令 $\dfrac{\sum_{i=1}^{k} s_i}{\sum_{i=1}^{\min\{m,n\}} s_i}$ 大于一定阈值

矩阵近似实验

低秩近似的应用之一：数据压缩

- 对于原$m \times n$矩阵**A**，直接存储则需记录mn个数字
- 若找到秩为k的近似矩阵**B**，最少需要记录$(m+n)k$个数字

用奇异值分解压缩以下图片：

矩阵近似实验

```
A = imread('originalfig.png');
A = rgb2gray(A);
A = double(A);
```

Variables – A

A

1162x1618 double

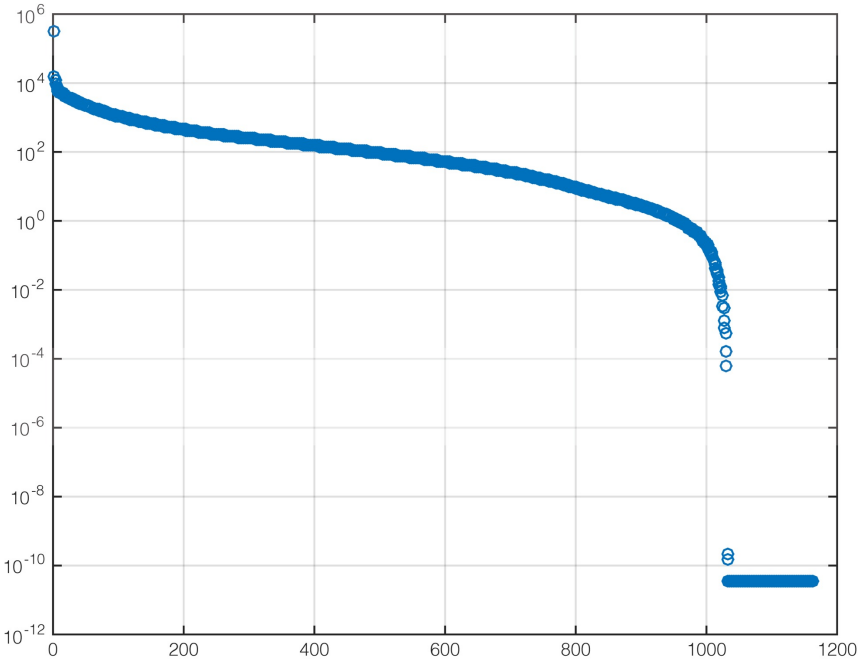
	1	2	3	4	5	6	7	8
1	255	255	255	255	255	255	255	255
2	255	255	255	255	255	255	254	255
3	254	255	255	255	255	255	255	255
4	255	255	255	255	255	255	255	255
5	255	254	255	255	255	255	255	255
6	255	254	255	255	255	254	254	255
7	255	255	255	255	255	255	255	255
8	255	255	255	254	255	255	255	255
9	255	255	254	255	255	255	255	255
10	255	255	255	255	255	255	254	255
11	255	255	255	254	255	255	255	255
12	255	255	254	255	254	255	255	255
13	255	255	255	255	255	255	255	255
14	254	255	255	255	255	255	255	255
15	255	255	255	255	254	255	255	255
16	255	255	255	255	255	255	255	255

把图片转成灰度图，每个像素取值0～255

0是黑色、255是白色

矩阵近似实验

```
A = imread('originalfig.png');
A = rgb2gray(A);
A = double(A);
[U S V] = svd(A);
figure
semilogy(diag(S), 'o');
grid on;
rank(double(A))
```



奇异值从大到小的取值

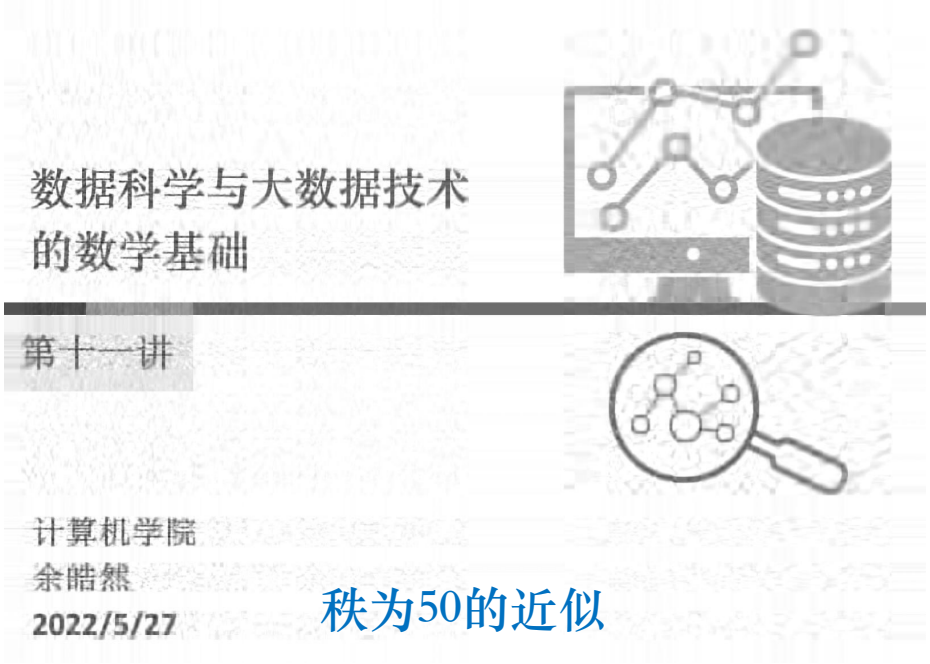
矩阵近似实验

```
A = imread('originalfig.png');
A = rgb2gray(A);
A = double(A);
[U S V] = svd(A);
figure
semilogy(diag(S), 'o');
grid on;
rank(double(A))
B = U(:,1:50)*S(1:50,1:50)*V(:,1:50)';
imwrite(uint8(B),'compressed1.jpg');
```

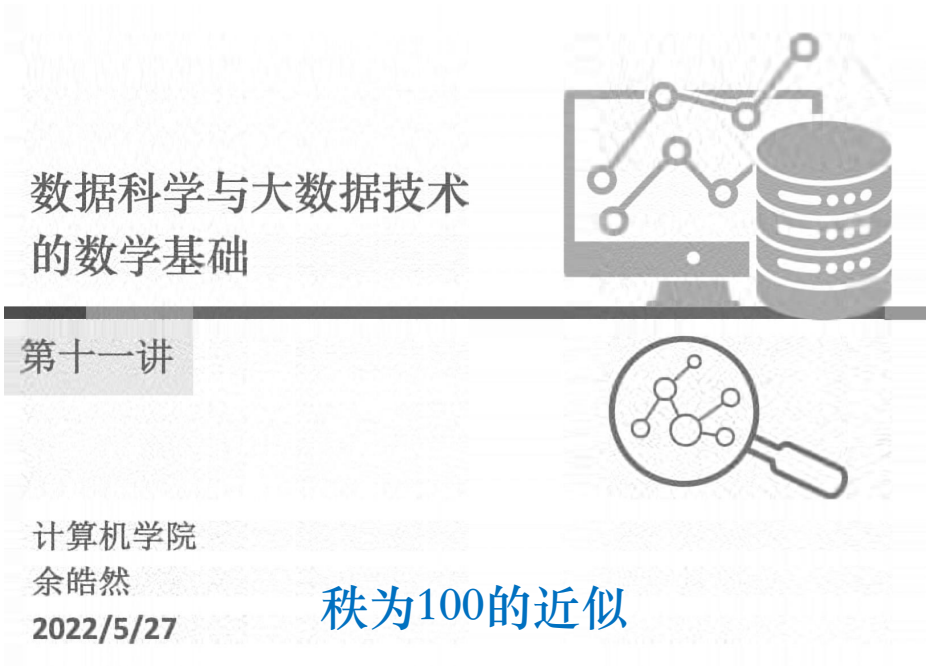
秩为50的近似

矩阵近似实验

```
A = imread('originalfig.png');
A = rgb2gray(A);
A = double(A);
[U S V] = svd(A);
figure
semilogy(diag(S), 'o');
grid on;
rank(double(A))
B = U(:,1:50)*S(1:50,1:50)*V(:,1:50)';
imwrite(uint8(B),'compressed1.jpg’);
B = U(:,1:100)*S(1:100,1:100)*V(:,1:100)';
imwrite(uint8(B),'compressed2.jpg');
```

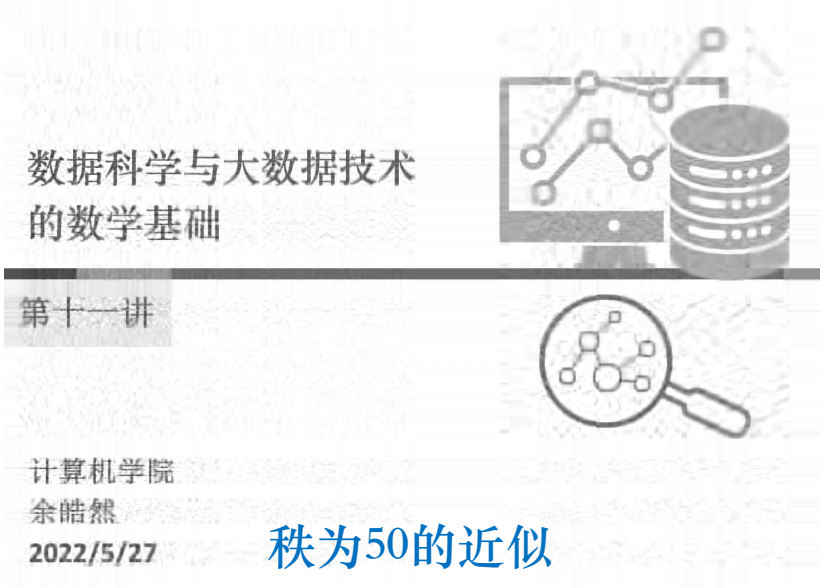


秩为50的近似

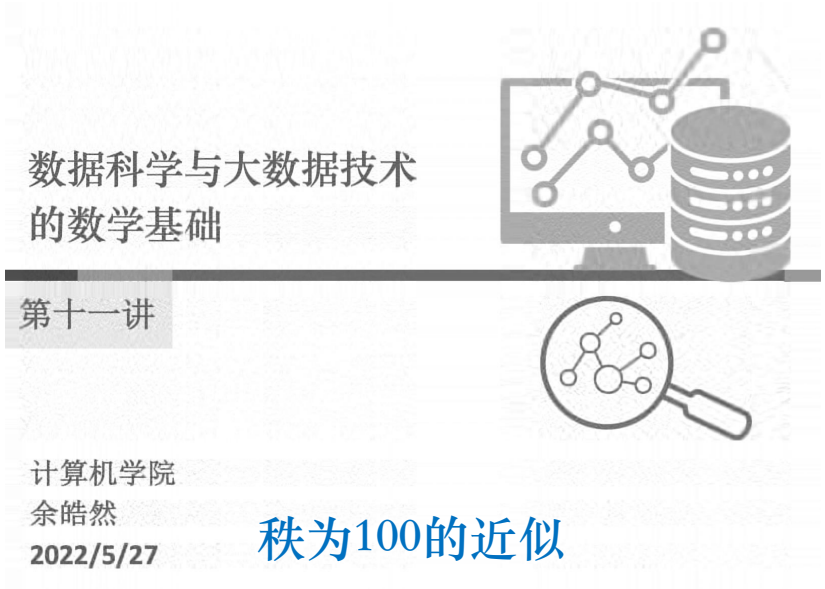


秩为100的近似

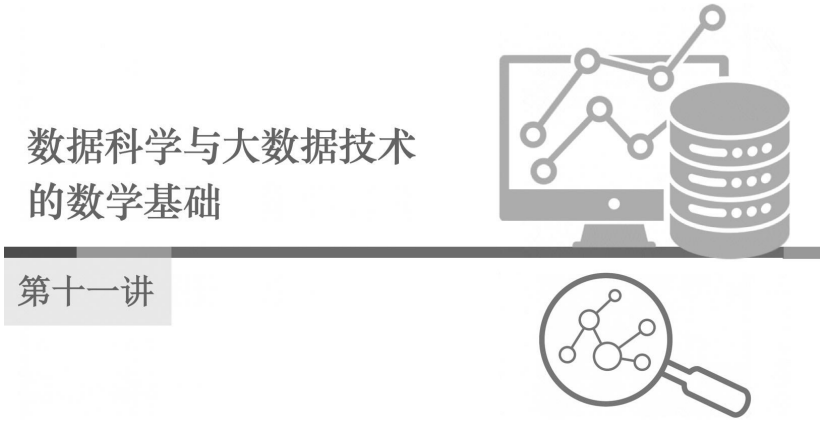
矩阵近似实验



秩为50的近似



秩为100的近似



秩为400的近似

奇异值分解

奇异值分解的应用：主成分分析

应用：主成分分析

回顾

主成分分析（Principal Component Analysis, PCA）

给定$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \mathbb{R}^n$及维数$k \geq 1$，求标准正交向量组$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k \in \mathbb{R}^n$从而最大化

$$\frac{1}{m}\sum_{i=1}^{m}\left(\langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle^2 + \cdots + \langle \boldsymbol{x}_i, \boldsymbol{v}_k\rangle^2\right).$$

将$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$分别近似为$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$的线性组合，即$\boldsymbol{x}_i \approx \sum_{j=1}^{k} a_{ij}\,\boldsymbol{v}_j, i = 1, \dots, m$。

主成分分析问题的最优解

最大化目标函数的$\boldsymbol{v}_1^*, \dots, \boldsymbol{v}_k^*$为矩阵$\mathbf{X}^T\mathbf{X}$的最大的$k$个特征值对应的$k$个单位特征向量。

基于幂迭代法的主成分分析：

- 用幂迭代法计算第一主成分（$\boldsymbol{u}_i=(\mathbf{X}^T\mathbf{X})^i\boldsymbol{u}_0$并标准化），结果记为$\boldsymbol{v}_1$
- 处理数据$\mathbf{X}$：$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle \boldsymbol{v}_1$
- 对新的$\mathbf{X}^T\mathbf{X}$计算前$k-1$个主成分

应用：主成分分析

回顾

主成分分析（Principal Component Analysis, PCA）

给定$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \mathbb{R}^n$及维数$k \geq 1$，求标准正交向量组$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k \in \mathbb{R}^n$从而最大化

$$\frac{1}{m}\sum_{i=1}^{m}\left(\langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle^2 + \cdots + \langle \boldsymbol{x}_i, \boldsymbol{v}_k\rangle^2\right).$$

将$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$分别近似为$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$的线性组合，即$\boldsymbol{x}_i \approx \sum_{j=1}^{k} a_{ij}\,\boldsymbol{v}_j, i = 1, \dots, m$。

主成分分析问题的最优解

最大化目标函数的$\boldsymbol{v}_1^*, \dots, \boldsymbol{v}_k^*$为矩阵$\mathbf{X}^T\mathbf{X}$的最大的$k$个特征值对应的$k$个单位特征向量。

基于幂迭代法的主成分分析：

- 用幂迭代法计算第一主成分（$\boldsymbol{u}_i=(\mathbf{X}^T\mathbf{X})^i\boldsymbol{u}_0$并标准化），结果记为$\boldsymbol{v}_1$
- 处理数据$\mathbf{X}$：$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle \boldsymbol{v}_1$
- 对新的$\mathbf{X}^T\mathbf{X}$计算前$k-1$个主成分

若可快速计算 **X** 的奇异值分解，如何对数据做主成分分析？

应用：主成分分析

回顾

主成分分析（Principal Component Analysis, PCA）

给定$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \mathbb{R}^n$及维数$k \geq 1$，求标准正交向量组$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k \in \mathbb{R}^n$从而最大化

$$\frac{1}{m}\sum_{i=1}^{m}\left(\langle \boldsymbol{x}_i, \boldsymbol{v}_1\rangle^2 + \cdots + \langle \boldsymbol{x}_i, \boldsymbol{v}_k\rangle^2\right).$$

将$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$分别近似为$\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$的线性组合，即$\boldsymbol{x}_i \approx \sum_{j=1}^{k} a_{ij}\,\boldsymbol{v}_j, i = 1, \dots, m$。

主成分分析问题的最优解

最大化目标函数的$\boldsymbol{v}_1^*, \dots, \boldsymbol{v}_k^*$为矩阵$\mathbf{X}^T\mathbf{X}$的最大的$k$个特征值对应的$k$个单位特征向量。

若$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$，有$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{S}^T\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T$

说明$\mathbf{V}$的列向量（即$\mathbf{V}^T$的行向量）为$\mathbf{X}^T\mathbf{X}$的单位特征向量

故只用取$\mathbf{V}$的前k个列向量即得到数据的k个主成分

应用：主成分分析

主成分计算方法

- 方法一：由幂迭代法逐个计算$\mathbf{X}^T\mathbf{X}$的k个主成分
- 方法二：对$\mathbf{X}$进行奇异值分解：$\mathbf{X}=\mathbf{USV}^T$，取$\mathbf{V}$的前$k$个列向量

选择哪一种方法?

- 方法一适用于k较小情况（如目标是数据可视化）
- 方法二可得到$\mathbf{X}^T\mathbf{X}$所有的特征向量，以及矩阵$\mathbf{U}$

$\mathbf{U}$的利用价值?

应用：主成分分析

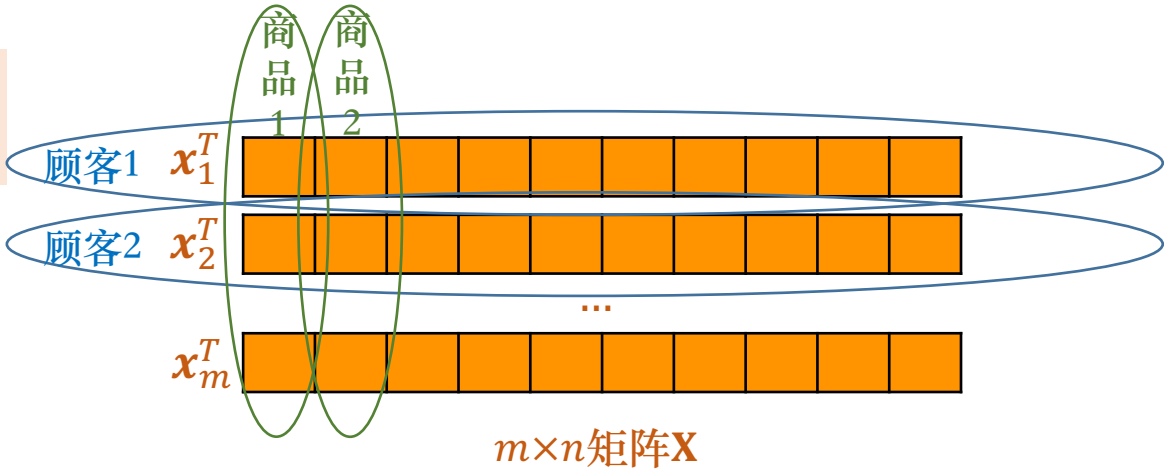
主成分计算方法

- 方法一：由幂迭代法逐个计算$\mathbf{X}^T\mathbf{X}$的k个主成分
- 方法二：对$\mathbf{X}$进行奇异值分解：$\mathbf{X}=\mathbf{USV}^T$，取$\mathbf{V}$的前$k$个列向量

选择哪一种方法?

- 方法一适用于k较小情况（如目标是数据可视化）
- 方法二可得到$\mathbf{X}^T\mathbf{X}$所有的特征向量，以及矩阵$\mathbf{U}$

若试图分析$\mathbf{X}$各列对应数据的主成分时，可利用矩阵$\mathbf{U}$



应用：主成分分析

SVD解决的低秩矩阵近似问题

给定$m \times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B} \in \mathbb{R}^{m \times n}} \|\mathbf{X} - \mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B}) \leq k.$$

计算$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_{i=1}^{\min\{m,n\}} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T$，可得$\mathbf{B}^* = \sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？

应用：主成分分析

SVD解决的低秩矩阵近似问题

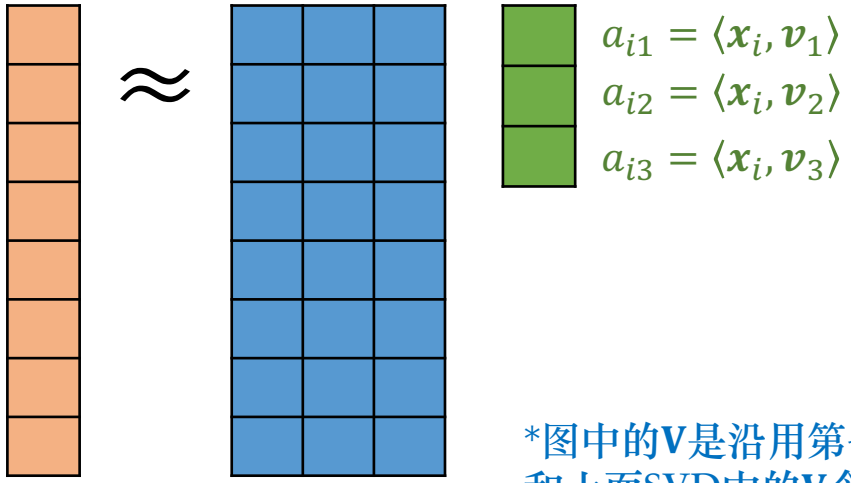
给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？

$$\boldsymbol{x}_i \approx \sum_{j=1}^{k} a_{ij}\,\boldsymbol{v}_j$$



*图中的$\mathbf{V}$是沿用第七讲中的符号，是$n\times k$矩阵
和上面SVD中的$\mathbf{V}$含义不同（SVD中$\mathbf{V}$是$n\times n$矩阵）

应用：主成分分析

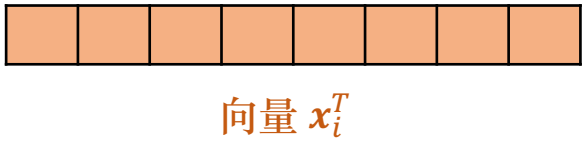
SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

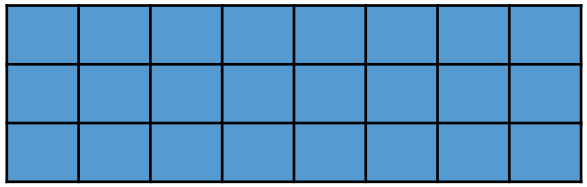
PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？



向量 $\boldsymbol{x}_i^T$

$\approx$

$\langle \boldsymbol{x}_i,\boldsymbol{v}_1\rangle, \langle \boldsymbol{x}_i,\boldsymbol{v}_2\rangle, \langle \boldsymbol{x}_i,\boldsymbol{v}_3\rangle$



矩阵$\mathbf{V}^T$

应用：主成分分析

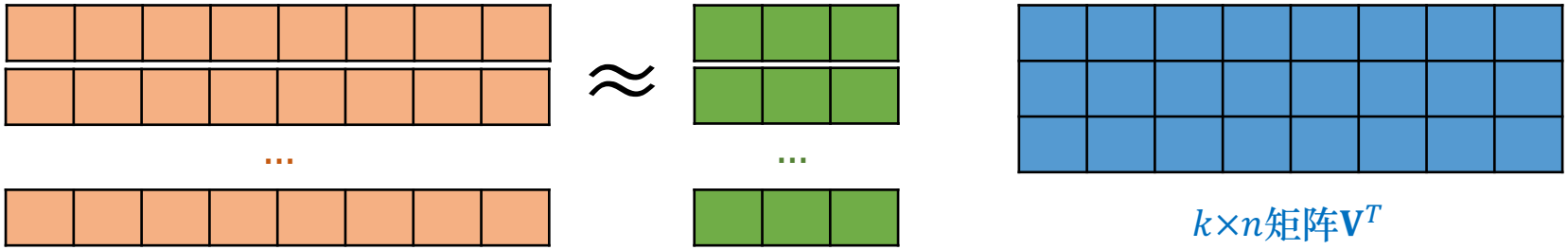
SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？



$m\times n$矩阵$\mathbf{X}$　　$k\times n$矩阵$\mathbf{V}^T$

$m\times k$矩阵，每行是$\langle \boldsymbol{x}_i,\boldsymbol{v}_1\rangle,\langle \boldsymbol{x}_i,\boldsymbol{v}_2\rangle,\langle \boldsymbol{x}_i,\boldsymbol{v}_k\rangle$

应用：主成分分析

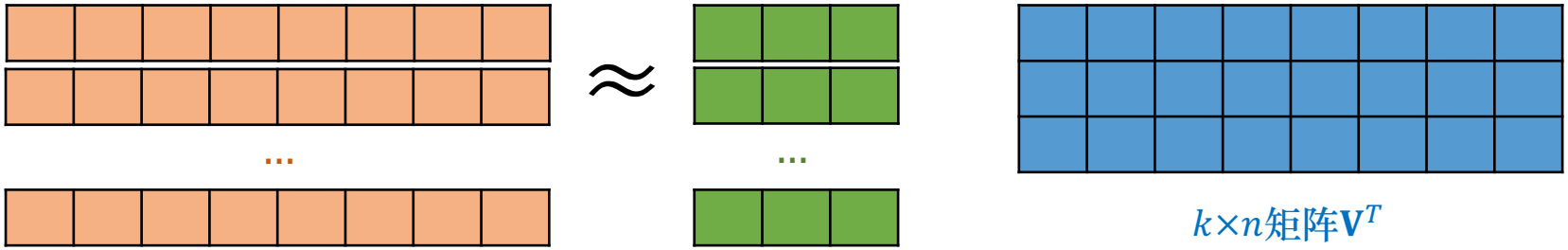
SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？



$m\times n$矩阵$\mathbf{X}$　　$k\times n$矩阵$\mathbf{V}^T$

$m\times k$矩阵，每行是$\langle \boldsymbol{x}_i,\boldsymbol{v}_1\rangle,\langle \boldsymbol{x}_i,\boldsymbol{v}_2\rangle,\langle \boldsymbol{x}_i,\boldsymbol{v}_k\rangle$

即矩阵$\mathbf{XV}$

应用：主成分分析

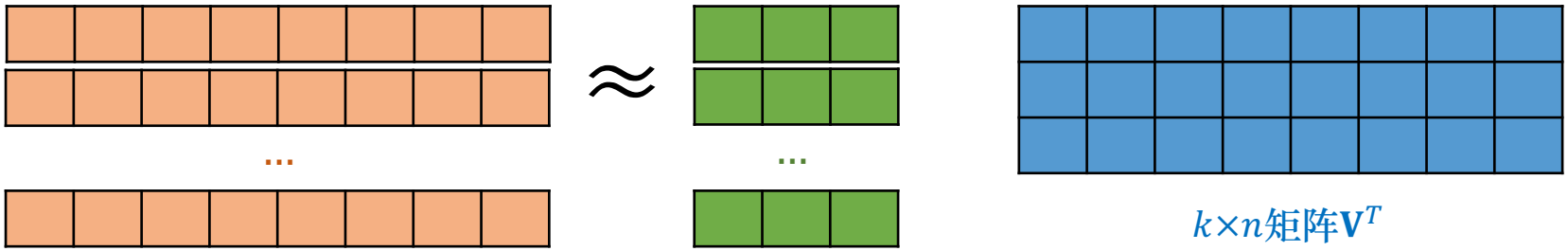
SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？



$m\times n$矩阵$\mathbf{X}$　　$m\times k$矩阵$\mathbf{XV}$　　$k\times n$矩阵$\mathbf{V}^T$

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{V}\in\mathbb{R}^{n\times k}} \|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F \quad \text{s.t. } \mathbf{V}\text{列向量构成标准正交向量组.}$$

应用：主成分分析

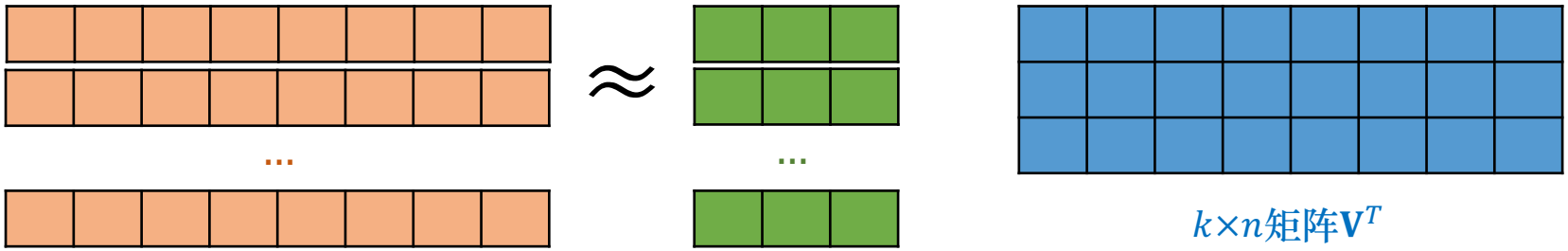
SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的问题？是否也可改写成最小化两个矩阵之间F-范数的形式？



$m\times n$矩阵$\mathbf{X}$　　$m\times k$矩阵$\mathbf{XV}$　　$k\times n$矩阵$\mathbf{V}^T$

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

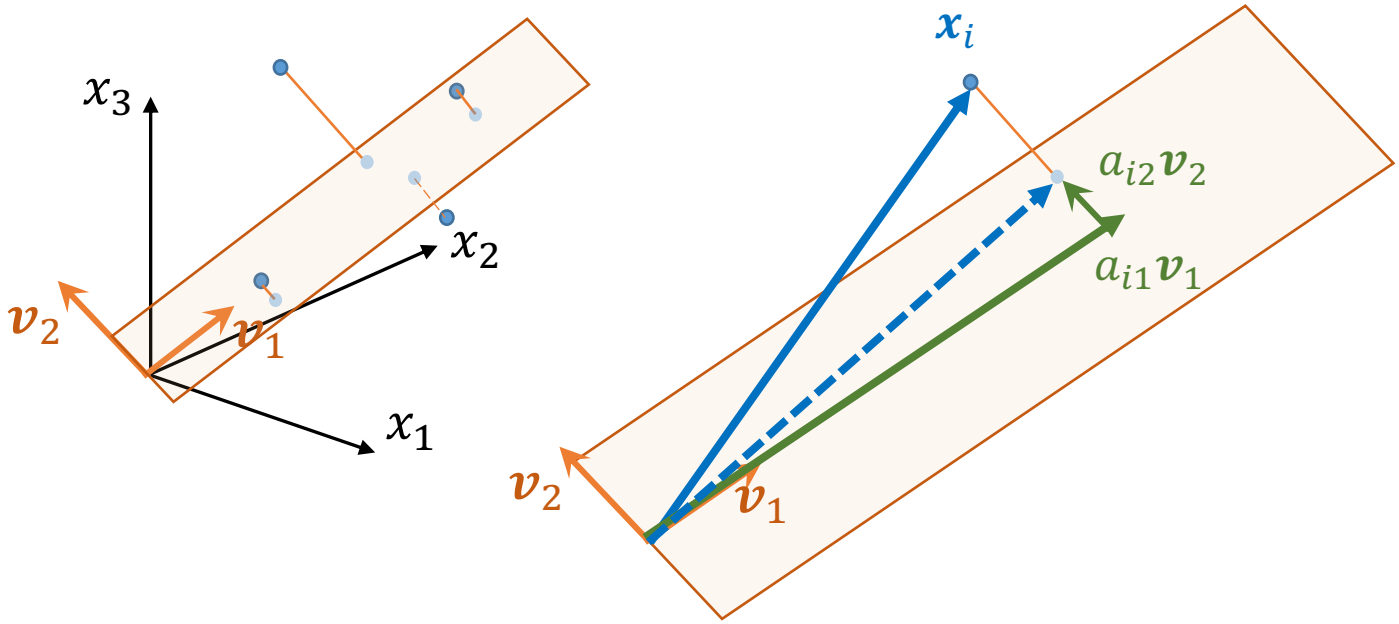
$$\min_{\mathbf{V}\in\mathbb{R}^{n\times k}} \|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F \quad \text{s.t. } \mathbf{V}\text{列向量构成标准正交向量组.}$$

易证最小化$\|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F$即等价于
第七讲中PCA的最小化问题
（用矩阵之差范数与行向量之差范数的关系）

应用：主成分分析

回顾

将（已完成预处理的）m个n维向量$\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \mathbb{R}^n$近似为$\sum_{j=1}^{k} a_{ij}\, \boldsymbol{v}_j, i = 1, \dots, m$。



$$\min \frac{1}{m} \sum_{i=1}^{m} (distance\ between\ \boldsymbol{x}_i\ and\ k-dimensional\ subspace\ spanned\ by\ \boldsymbol{v}_1, \dots, \boldsymbol{v}_k)^2$$

$$\min \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{x}_i\|_2^2 - \frac{1}{m} \sum_{i=1}^{m} (\langle \boldsymbol{x}_i, \boldsymbol{v}_1 \rangle^2 + \cdots + \langle \boldsymbol{x}_i, \boldsymbol{v}_k \rangle^2)$$

应用：主成分分析

SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{V}\in\mathbb{R}^{n\times k}} \|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F \quad \text{s.t. } \mathbf{V}\text{列向量构成标准正交向量组.}$$

应用：主成分分析

SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{V}\in\mathbb{R}^{n\times k}} \|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F \quad \text{s.t. } \mathbf{V}\text{列向量构成标准正交向量组.}$$

以上两个矩阵近似问题得到的近似矩阵是否一样？

$$\mathbf{B}^*=\mathbf{X}\mathbf{V}^*(\mathbf{V}^*)^T\ ?$$

* 这里$\mathbf{V}^*$是PCA问题中经优化的$n\times k$矩阵
和上面SVD中的$\mathbf{V}$含义不同（SVD中$\mathbf{V}$是$n\times n$矩阵）

应用：主成分分析

SVD解决的低秩矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{X}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

计算$\mathbf{X}=\mathbf{U}\mathbf{S}\mathbf{V}^T=\sum_{i=1}^{\min\{m,n\}} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$，可得$\mathbf{B}^*=\sum_{i=1}^{k} s_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

PCA解决的矩阵近似问题

给定$m\times n$矩阵$\mathbf{X}$及正整数k，解决如下问题：

$$\min_{\mathbf{V}\in\mathbb{R}^{n\times k}} \|\mathbf{X}-\mathbf{X}\mathbf{V}\mathbf{V}^T\|_F \quad \text{s.t. } \mathbf{V}\text{列向量构成标准正交向量组.}$$

以上两个矩阵近似问题得到的近似矩阵一样：$\mathbf{B}^*=\mathbf{X}\mathbf{V}^*(\mathbf{V}^*)^T$

证明方法一：取$\mathbf{V}^*$为$\mathbf{U}\mathbf{S}\mathbf{V}^T$中$\mathbf{V}$的前$k$列、将$\mathbf{X}$展成$\mathbf{U}\mathbf{S}\mathbf{V}^T$；证$\mathbf{X}\mathbf{V}^*(\mathbf{V}^*)^T$最多有$k$个非零奇异值

证明方法二：利用$\boldsymbol{u}_i=\frac{\mathbf{X}\boldsymbol{v}_i}{s_i}$化简$\mathbf{B}^*=\sum_{i=1}^{k}\mathbf{X}\boldsymbol{v}_i\boldsymbol{v}_i^T$；和$\mathbf{X}\mathbf{V}^*(\mathbf{V}^*)^T$对比

说明实际上PCA通过k个主成分得到的近似矩阵就是$\mathbf{X}$的k秩近似矩阵

奇异值分解

奇异值分解的应用：矩阵补全

应用：矩阵补全

问题：对矩阵A仅知道部分元素取值，如何估测其余元素取值（即补全矩阵）？

$$\begin{bmatrix} 2 & 5 & ? & 1 & 2 & ? \\ ? & ? & 2 & ? & 3 & ? \\ ? & ? & 1 & ? & ? & 2 \\ ? & 4 & ? & ? & 1 & 5 \\ 1 & ? & ? & ? & ? & 2 \\ ? & 4 & ? & ? & ? & ? \\ ? & 1 & 1 & 1 & ? & ? \\ 2 & ? & ? & 4 & 5 & ? \\ 1 & 3 & 2 & ? & 4 & 5 \end{bmatrix}$$

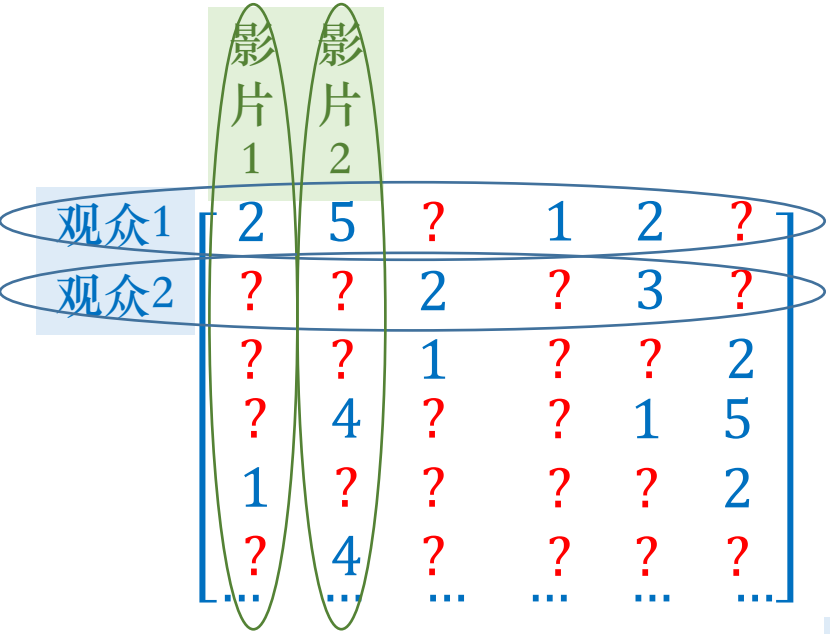
应用：矩阵补全

应用场景：推荐系统

2006年10月2日，Netflix开启比赛

应用：矩阵补全

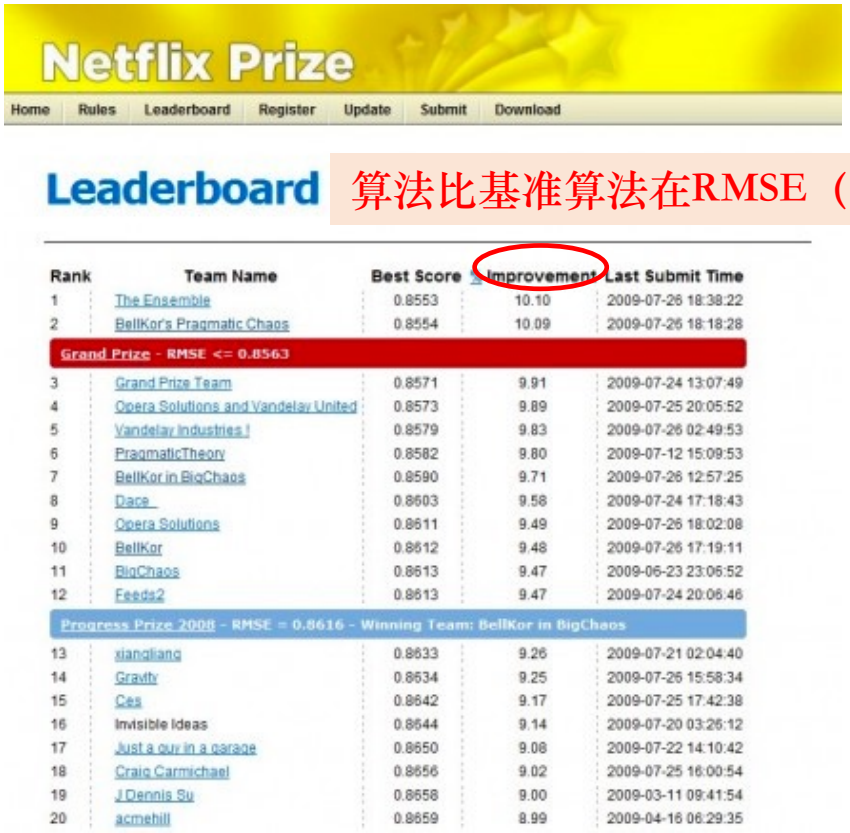
Netflix比赛题目：给定480,189个观众对17,770部电影的100,480,507个评分，估测观众对电影的评分



即矩阵有480,189行、17,770列，100,480,507个元素的值已知（约99%的元素取值未知），要求补全矩阵从而最小化补全值（即估测值）与真实值之间的欧氏距离

若可准确补全矩阵，即可准确估测观众对没看过的电影的喜好程度，据此进行推荐

应用：矩阵补全



2006.10.2~2009.7.26，不断有新的算法刷新记录

（2010.3.12 Netflix因隐私条例停办比赛）

应用：矩阵补全

Netflix Prize

Home | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

算法比基准算法在RMSE（Rooted Mean Squared Error/均方根误差）方面的提升

Rank	Team Name	Best Score	Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28
Grand Prize - RMSE <= 0.8563				
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52
5	Vandelay Industries !	0.8579	9.83	2009-07-26 02:49:53
6	PragmaticTheory	0.8582	9.80	2009-07-12 15:09:53
7	BellKor in BigChaos	0.8590	9.71	2009-07-26 12:57:25
8	Dace	0.8603	9.58	2009-07-24 17:18:43
9	Opera Solutions	0.8611	9.49	2009-07-26 18:02:08
10	BellKor	0.8612	9.48	2009-07-26 17:19:11
11	BigChaos	0.8613	9.47	2009-06-23 23:06:52
12	Feeds2	0.8613	9.47	2009-07-24 20:06:46
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8633	9.26	2009-07-21 02:04:40
14	Gravity	0.8634	9.25	2009-07-26 15:58:34
15	Ces	0.8642	9.17	2009-07-25 17:42:38
16	Invisible Ideas	0.8644	9.14	2009-07-20 03:26:12
17	Just a guy in a garage	0.8650	9.08	2009-07-22 14:10:42
18	Craig Carmichael	0.8656	9.02	2009-07-25 16:00:54
19	J Dennis Su	0.8658	9.00	2009-03-11 09:41:54
20	acmehill	0.8659	8.99	2009-04-16 06:29:35

2006.10.2~2009.7.26，不断有新的算法刷新记录 在测试集（非验证集）上效果最好（Test RMSE=0.8567）的队伍获冠军

（2010.3.12 Netflix因隐私条例停办比赛）

图片取自 Dan Jackson <The Netflix Prize: How a $1 Million Contest Changed Binge-Watching Forever>

应用：矩阵补全

The BigChaos Solution to the Netflix Grand Prize

Andreas Töscher and Michael Jahrer

commendo research & consulting
Neuer Weg 23, A-8580 Köflach, Austria
{*andreas.toescher,michael.jahrer*}*@commendo.at*

Robert M. Bell*

AT&T Labs - Research
Florham Park, NJ

September 5, 2009

冠军方案论文

The other main driving force in the competition was the ensemble idea. The ensemble idea was part of the competition from the beginning and evolved over time. In the beginning, we used different models with different parametrization and a linear blending. The models were trained individually and the meta parameters got optimized to reduce the RMSE of the individual model. The linear blend was replaced by a nonlinear one, a neural network. This was basically the solution for the progress prize 2008, a ensemble of independently trained and tuned predictors, and a neural network for the blending. In fall 2008, we realized that training and optimizing the predictors individually is not optimal. Best blending results are achieved when the whole ensemble has the right tradeoff between diversity and accuracy. So we started to train the predictors sequentially and stopped the training when the blending improvement was best. Also the meta parameters were tuned, to achieve best blending performance. The next step in the evolution of the ensemble idea was to replace the single neural network blend by an ensemble of blends. In order to maximize diversity within the blending ensemble, the blends used different subsets of predictors and different blending methods. Figure 6 shows the RMSE improvements compared to the number of predictors. Within the first predictors there are a lot of different blends (the first 18 are listed in Appendix C). This clearly shows that the diverse set of nonlinear probe blends is an important part of our solution.

方案的思想之一：集成学习

应用：矩阵补全

如何将矩阵补全问题改写成低秩矩阵近似问题?

电影

观众

$$\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 1 \end{bmatrix}$$

应用：矩阵补全

如何将矩阵补全问题改写成低秩矩阵近似问题?

电影

观众 $\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 1 \end{bmatrix}$

猜测：矩阵的秩为1
（即观众对电影的喜好排序一致、评分向量线性相关）

$$\begin{bmatrix} 2 & 2 & 1 & 1 \\ 4 & 4 & 2 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}$$

估计的矩阵秩为1而且在已知评分的位置的元素和原矩阵相同

应用：矩阵补全

如何将矩阵补全问题改写成低秩矩阵近似问题?

电影　　若给定矩阵修改为

观众

$$\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 3 \end{bmatrix}$$

猜测：矩阵的秩为1
（即观众对电影的喜好排序一致、评分向量线性相关）

$$\begin{bmatrix} 2 & 2 & 1 & 1 \\ 4 & 4 & 2 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}$$

估计的矩阵秩为1而且在已知评分的位置的元素和原矩阵相同

应用：矩阵补全

如何将矩阵补全问题改写成低秩矩阵近似问题?

电影　　若给定矩阵修改为

观众

$$\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 3 \end{bmatrix}$$

猜测：矩阵的秩为1

（即观众对电影的喜好排序一致、评分向量线性相关）

$$\begin{bmatrix} 2 & 2 & 1 & 1 \\ 4 & 4 & 2 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}$$

（在已知评分上）有近似误差

应用：矩阵补全

如何将矩阵补全问题改写成低秩矩阵近似问题?

电影　　若给定矩阵修改为

观众 $\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 3 \end{bmatrix}$ 矩阵A

猜测：矩阵的秩为1
（即观众对电影的喜好排序一致、评分向量线性相关）

$\begin{bmatrix} 2 & 2 & 1 & 1 \\ 4 & 4 & 2 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}$ 矩阵C

（在已知评分上）有近似误差

低秩矩阵补全（Low-Rank Matrix Completion）

给定$m \times n$矩阵A及正整数k，解决如下问题：

$$\min_{C \in \mathbb{R}^{m \times n}} \sum_{\text{observed}(i,j)} (A_{ij} - C_{ij})^2 \quad \text{s.t. } \operatorname{rank}(C) = k.$$

信息不全

应用：矩阵补全

回顾

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

利用SVD对矩阵$\mathbf{A}$进行低秩近似的步骤：

（1）对矩阵$\mathbf{A}$做奇异值分解；

（2）计算矩阵$\mathbf{B}=\sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T$。

低秩矩阵补全（Low-Rank Matrix Completion）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{C}\in\mathbb{R}^{m\times n}} \sum_{\text{observed}(i,j)} \left(\mathrm{A}_{ij}-\mathrm{C}_{ij}\right)^2 \quad \text{s.t. } \operatorname{rank}(\mathbf{C})=k.$$

如何解决？

应用：矩阵补全

回顾

低秩矩阵近似（Low-Rank Matrix Approximation）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{B}\in\mathbb{R}^{m\times n}} \|\mathbf{A}-\mathbf{B}\|_F \quad \text{s.t. } \operatorname{rank}(\mathbf{B})\le k.$$

利用SVD对矩阵$\mathbf{A}$进行低秩近似的步骤：

（1）对矩阵$\mathbf{A}$做奇异值分解；

（2）计算矩阵$\mathbf{B}=\sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T$。

例

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

低秩矩阵补全（Low-Rank Matrix Completion）

给定$m\times n$矩阵$\mathbf{A}$及正整数k，解决如下问题：

$$\min_{\mathbf{C}\in\mathbb{R}^{m\times n}} \sum_{\text{observed}(i,j)} (\mathrm{A}_{ij}-\mathrm{C}_{ij})^2 \quad \text{s.t. } \operatorname{rank}(\mathbf{C})=k.$$

例

$$\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 2 \end{bmatrix}$$

如何解决？

应用：矩阵补全

例

$$\begin{bmatrix} 2 & ? & ? & 1 \\ ? & 4 & ? & 2 \\ ? & 2 & 1 & 2 \end{bmatrix}$$

[PDF] Matrix completion and low-rank SVD via fast alternating least squares

T Hastie, R Mazumder, JD Lee, R Zadeh - The Journal of Machine Learning ..., 2015 - jmlr.org

The matrix-completion problem has attracted a lot of attention, largely as a result of the celebrated Netflix competition. Two popular approaches for solving the problem are nuclear-norm-regularized matrix approximation (Candes and Tao, 2009; Mazumder et al., 2010), and maximum-margin matrix factorization (Srebro et al., 2005). These two procedures are in some cases solving equivalent problems, but with quite different algorithms. In this article we bring the two approaches together, leading to an efficient algorithm for large matrix ...

☆ Save 99 Cite Cited by 403 Related articles All 29 versions »

1. Replace the missing entries in X with the corresponding entries from the current estimate $\widehat{M}$:

$$\widehat{X} \leftarrow P_\Omega(X) + P_\Omega^\perp(\widehat{M}); \tag{2}$$

2. Update $\widehat{M}$ by computing the soft-thresholded SVD of $\widehat{X}$:

$$\widehat{X} = UDV^T \tag{3}$$
$$\widehat{M} \leftarrow US_\lambda(D)V^T, \tag{4}$$

where the soft-thresholding operator S_λ operates element-wise on the diagonal matrix D, and replaces D_{ii} with $(D_{ii} - \lambda)_+$. With large λ many of the diagonal elements will be set to zero, leading to a low-rank solution for Problem (1).

论文对Mazumder et al. (2010)方法的总结

奇异值分解

奇异值分解的应用：潜在语义分析

应用：潜在语义分析

潜在语义分析（Latent Semantic Analysis）属于自然语言处理，旨在通过矩阵分解等发掘文本与词语之间基于话题的语义关系

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

如果搜索“dies, dagger”，该如何根据相关性排列这五个文本?

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

潜在语义分析（Latent Semantic Analysis）属于自然语言处理，旨在通过矩阵分解等发掘文本与词语之间基于话题的语义关系

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

如果搜索“dies, dagger”，该如何根据相关性排列这五个文本?

猜测： d3最前（同时包含两个搜索词），d2和d4次之

对于d1和d5?

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

潜在语义分析（Latent Semantic Analysis）属于自然语言处理，旨在通过矩阵分解等发掘文本与词语之间基于话题的语义关系

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

如果搜索“dies, dagger”，该如何根据相关性排列这五个文本?

猜测： d3最前（同时包含两个搜索词），d2和d4次之

d1通过Romeo与d3关联、进而与“dies”和“dagger”关联；通过Juliet与d2关联、进而与“dagger”关联

d5通过New-Hampshire与d4关联、进而与“dies”关联

例子取自Alex Thomo <SENG 474: Data Mining>

若把d1、d5、“dies, dagger”分别近似成向量，是否可以见到d1向量与“dies, dagger”向量更接近?

应用：潜在语义分析

潜在语义分析（Latent Semantic Analysis）属于自然语言处理，旨在通过矩阵分解等发掘文本与词语之间基于话题的语义关系

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

如果搜索“dies, dagger”，该如何根据相关性排列这五个文本?

猜测： d3最前（同时包含两个搜索词），d2和d4次之

d1通过Romeo与d3关联、进而与“dies”和“dagger”关联；通过Juliet与d2关联、进而与“dagger”关联

d5通过New-Hampshire与d4关联、进而与“dies”关联

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

得到8×5矩阵**A**

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: "Live free or die", that's the New-Hampshire's motto.
d5: Did you know, New-Hampshire is in New-England.

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

得到8×5矩阵**A**

为方便结果可视化，将矩阵近似为秩为2的矩阵：

（1）做奇异值分解：$\mathbf{A} = \mathbf{USV}^T$

（2）对S的对角线元素，只保留前2位，其余置零

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

输入矩阵

```
>> [U, S, V]=svd(ans)

U =

   -0.3962    0.2801   -0.5712    0.4497   -0.1018   -0.0781    0.2799    0.3760
   -0.3143    0.4495    0.4106    0.5130    0.2039    0.0781   -0.2799   -0.3760
   -0.1782    0.2690    0.4973   -0.2570    0.0431    0.4145    0.2432    0.5914
   -0.4384    0.3685    0.0129   -0.5773   -0.2196   -0.4925    0.0367   -0.2153
   -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.4915   -0.4041    0.4561
   -0.5240   -0.2464   -0.3387   -0.2728    0.1548    0.5706   -0.3166   -0.1607
   -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.0791    0.7207   -0.2954
   -0.3264   -0.4597    0.3170    0.2372   -0.7249    0.0000    0.0000         0


S =

    2.2853         0         0         0         0
         0    2.0103         0         0         0
         0         0    1.3607         0         0
         0         0         0    1.1181         0
         0         0         0         0    0.7966
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0


V =

   -0.3109    0.3629   -0.1180    0.8610    0.1281
   -0.4073    0.5407    0.6767   -0.2874    0.0343
   -0.5945    0.2001   -0.6592   -0.3582   -0.2093
   -0.6030   -0.6954    0.1984    0.0531    0.3326
   -0.1428   -0.2287    0.2330    0.2122   -0.9100
```

MATLAB计算奇异值分解

应用：潜在语义分析

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

输入矩阵

```
>> [U, S, V]=svd(ans)

U =

   -0.3962    0.2801   -0.5712    0.4497   -0.1018   -0.0781    0.2799    0.3760
   -0.3143    0.4495    0.4106    0.5130    0.2039    0.0781   -0.2799   -0.3760
   -0.1782    0.2690    0.4973   -0.2570    0.0431    0.4145    0.2432    0.5914
   -0.4384    0.3685    0.0129   -0.5773   -0.2196   -0.4925    0.0367   -0.2153
   -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.4915   -0.4041    0.4561
   -0.5240   -0.2464   -0.3387   -0.2728    0.1548    0.5706   -0.3166   -0.1607
   -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.0791    0.7207   -0.2954
   -0.3264   -0.4597    0.3170    0.2372   -0.7249    0.0000    0.0000         0

S =

    2.2853         0         0         0         0
         0    2.0103         0         0         0
         0         0    1.3607         0         0
         0         0         0    1.1181         0
         0         0         0         0    0.7966
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0

V =

   -0.3109    0.3629   -0.1180    0.8610    0.1281
   -0.4073    0.5407    0.6767   -0.2874    0.0343
   -0.5945    0.2001   -0.6592   -0.3582   -0.2093
   -0.6030   -0.6954    0.1984    0.0531    0.3326
   -0.1428   -0.2287    0.2330    0.2122   -0.9100
```

MATLAB计算奇异值分解

应用：潜在语义分析

采用的是第二种SVD展开方式

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

```
>> [U, S, V]=svd(ans)

U =

  -0.3962    0.2801   -0.5712    0.4497   -0.1018   -0.0781    0.2799    0.3760
  -0.3143    0.4495    0.4106    0.5130    0.2039    0.0781   -0.2799   -0.3760
  -0.1782    0.2690    0.4973   -0.2570    0.0431    0.4145    0.2432    0.5914
  -0.4384    0.3685    0.0129   -0.5773   -0.2196   -0.4925    0.0367   -0.2153
  -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.4915   -0.4041    0.4561
  -0.5240   -0.2464   -0.3387   -0.2728    0.1548    0.5706   -0.3166   -0.1607
  -0.2639   -0.3459    0.1458    0.0475    0.4175   -0.0791    0.7207   -0.2954
  -0.3264   -0.4597    0.3170    0.2372   -0.7249    0.0000    0.0000         0

S =

   2.2853         0         0         0         0
        0    2.0103         0         0         0
        0         0    1.3607         0         0
        0         0         0    1.1181         0
        0         0         0         0    0.7966
        0         0         0         0         0
        0         0         0         0         0
        0         0         0         0         0

V =

  -0.3109    0.3629   -0.1180    0.8610    0.1281
  -0.4073    0.5407    0.6767   -0.2874    0.0343
  -0.5945    0.2001   -0.6592   -0.3582   -0.2093
  -0.6030   -0.6954    0.1984    0.0531    0.3326
  -0.1428   -0.2287    0.2330    0.2122   -0.9100
```

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

采用的是第二种SVD展开方式

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$$\approx \begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

两个行向量为单位向量且正交

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.905 & 0.563 \\ -0.717 & 0.905 \\ -0.407 & 0.541 \\ -1.001 & 0.742 \\ -0.603 & -0.695 \\ -1.197 & -0.494 \\ -0.603 & -0.695 \\ -0.745 & -0.925 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

两个行向量为单位向量且正交

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.905 & 0.563 \\ -0.717 & 0.905 \\ -0.407 & 0.541 \\ -1.001 & 0.742 \\ -0.603 & -0.695 \\ -1.197 & -0.494 \\ -0.603 & -0.695 \\ -0.745 & -0.925 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

“dagger”近似表示为[−1.001,0.742]

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

两个行向量为单位向量且正交

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.905 & 0.563 \\ -0.717 & 0.905 \\ -0.407 & 0.541 \\ -1.001 & 0.742 \\ -0.603 & -0.695 \\ -1.197 & -0.494 \\ -0.603 & -0.695 \\ -0.745 & -0.925 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

“dies, dagger”近似表示为 $\left[\frac{-1.001-1.197}{2}, \frac{0.742-0.494}{2}\right] = [-1.099, 0.124]$

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

两个列向量为单位向量且正交

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} -0.711 & -0.930 & -1.357 & -1.378 & -0.327 \\ 0.730 & 1.087 & 0.402 & -1.397 & -0.460 \end{bmatrix}$$

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

两个列向量为单位向量且正交

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix} \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

8×5矩阵　　8×2矩阵　　2×2矩阵　　2×5矩阵

将d1、d5、“dies, dagger”分别近似成2维向量，再于2维平面对比相似度

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

$\approx$

$$\begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix} \begin{bmatrix} -0.711 & -0.930 & -1.357 & -1.378 & -0.327 \\ 0.730 & 1.087 & 0.402 & -1.397 & -0.460 \end{bmatrix}$$

d1近似表示为[−0.711,0.730]，d5近似表示为[−0.327, −0.460]

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

若把d1、d5、“dies, dagger”分别近似成向量，是否可以见到d1向量与“dies, dagger”向量更接近?

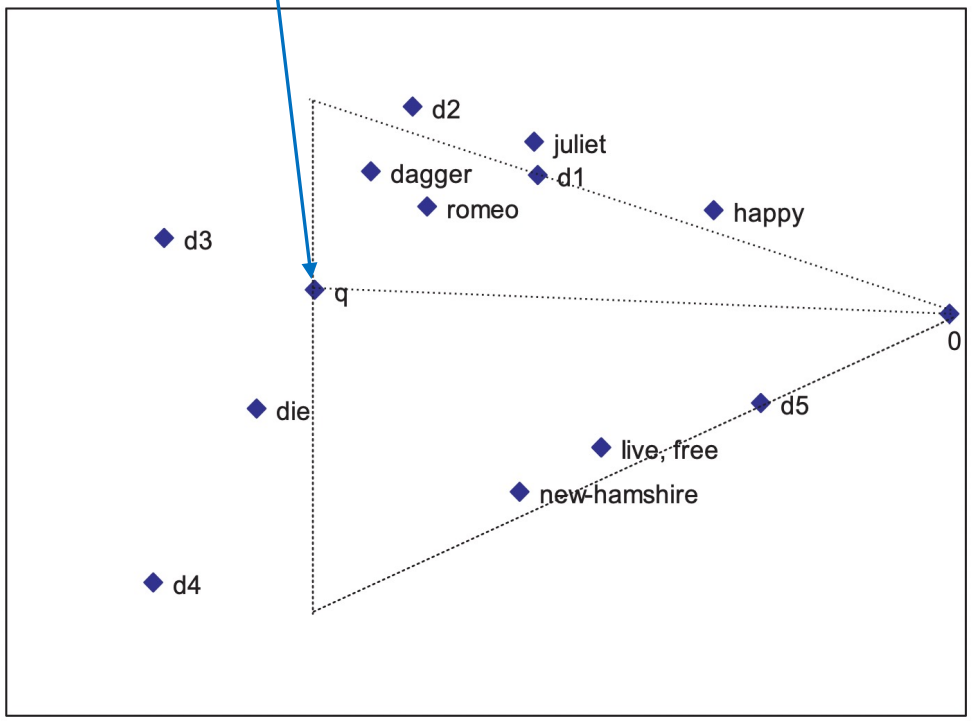
“dies, dagger”近似表示为[−1.099,0.124]
d1近似表示为[−0.711,0.730]，d5近似表示为[−0.327, −0.460]

可通过计算向量夹角判断相似度

例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

“dies, dagger”近似表示为[−1.099,0.124]
d1近似表示为[−0.711,0.730]，d5近似表示为[−0.327, −0.460]



d1与q（即“dies, dagger”）的夹角 比 d5与q（即“dies, dagger”）的夹角 小

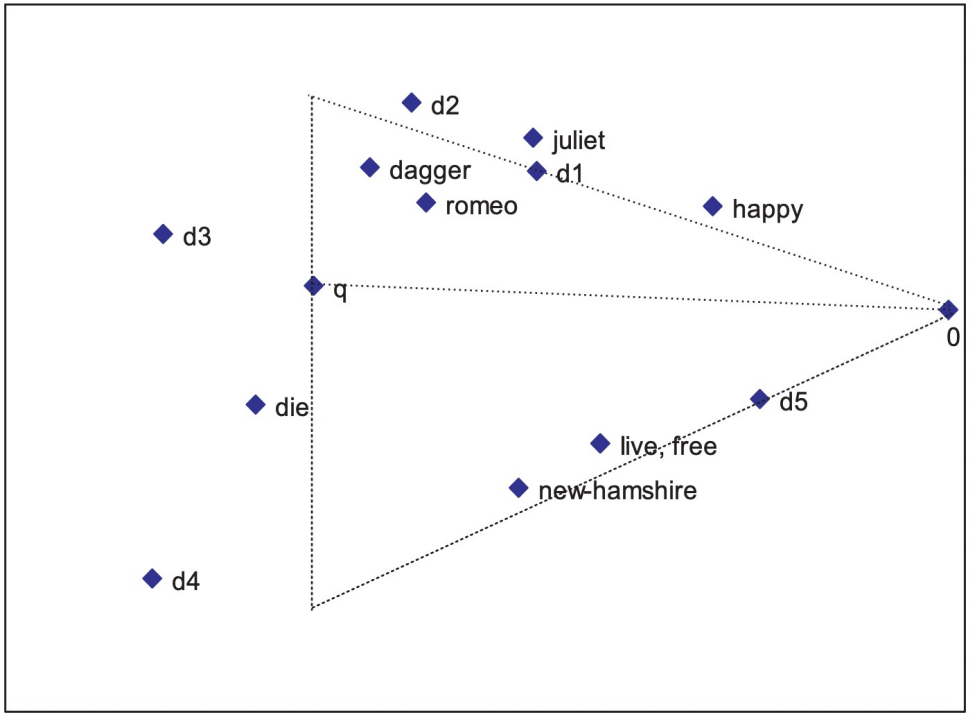
例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

d1: Romeo and Juliet.
d2: Juliet: O happy dagger!
d3: Romeo died by dagger.
d4: “Live free or die”, that’s the New-Hampshire’s motto.
d5: Did you know, New-Hampshire is in New-England.

如果搜索“dies, dagger”，该如何根据相关性排列这五个文本?

d1比d2的排序更前



例子取自Alex Thomo <SENG 474: Data Mining>

应用：潜在语义分析

data	1	2	1	5	0	0	0
info	1	2	1	5	0	0	0
retrieval	1	2	1	5	0	0	0
brain	0	0	0	0	2	3	1
lung	0	0	0	0	2	3	1

例子取自Duen Horng Chau <Georgia Tech CSE6242: Latent Semantic Indexing>

应用：潜在语义分析

```
>> [U S V] = svd(A)

U =

   -0.5774         0   -0.8165         0         0
   -0.5774         0    0.4082         0   -0.7071
   -0.5774         0    0.4082         0    0.7071
         0   -0.7071         0    0.7071         0
         0   -0.7071         0   -0.7071         0


S =

    9.6437         0         0         0         0         0         0
         0    5.2915         0         0         0         0         0
         0         0    0.0000         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0


V =

   -0.1796         0   -0.8980    0.1920   -0.1796    0.2880    0.0960
   -0.3592         0   -0.2735   -0.4761   -0.0547   -0.7141   -0.2380
   -0.1796         0   -0.1367    0.0292    0.9727    0.0439    0.0146
   -0.8980         0    0.3163    0.1462   -0.1367    0.2193    0.0731
         0   -0.5345         0    0.7143         0   -0.4286   -0.1429
         0   -0.8018         0   -0.4286         0    0.3571   -0.2143
         0   -0.2673         0   -0.1429         0   -0.2143    0.9286
```

$$
\begin{array}{l}
\text{data} \\ \text{info} \\ \text{retrieval} \\ \text{brain} \\ \text{lung}
\end{array}
\begin{bmatrix}
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 3 & 1 \\
0 & 0 & 0 & 0 & 2 & 3 & 1
\end{bmatrix}
$$

$$
\approx
\begin{bmatrix}
0.58 & 0 \\
0.58 & 0 \\
0.58 & 0 \\
0 & 0.71 \\
0 & 0.71
\end{bmatrix}
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\begin{bmatrix}
0.18 & 0.36 & 0.18 & 0.90 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.53 & 0.80 & 0.27
\end{bmatrix}
$$

例子取自Duen Horng Chau <Georgia Tech CSE6242: Latent Semantic Indexing>

应用：潜在语义分析

```
>> [U S V] = svd(A)

U =

   -0.5774         0   -0.8165         0         0
   -0.5774         0    0.4082         0   -0.7071
   -0.5774         0    0.4082         0    0.7071
         0   -0.7071         0    0.7071         0
         0   -0.7071         0   -0.7071         0


S =

    9.6437         0         0         0         0         0         0
         0    5.2915         0         0         0         0         0
         0         0    0.0000         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0


V =

   -0.1796         0   -0.8980    0.1920   -0.1796    0.2880    0.0960
   -0.3592         0   -0.2735   -0.4761   -0.0547   -0.7141   -0.2380
   -0.1796         0   -0.1367    0.0292    0.9727    0.0439    0.0146
   -0.8980         0    0.3163    0.1462   -0.1367    0.2193    0.0731
         0   -0.5345         0    0.7143         0   -0.4286   -0.1429
         0   -0.8018         0   -0.4286         0    0.3571   -0.2143
         0   -0.2673         0   -0.1429         0   -0.2143    0.9286
```

$$
\begin{array}{l}
\text{data} \\ \text{info} \\ \text{retrieval} \\ \text{brain} \\ \text{lung}
\end{array}
\begin{bmatrix}
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
1 & 2 & 1 & 5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 3 & 1 \\
0 & 0 & 0 & 0 & 2 & 3 & 1
\end{bmatrix}
\approx
\begin{bmatrix}
0.58 & 0 \\
0.58 & 0 \\
0.58 & 0 \\
0 & 0.71 \\
0 & 0.71
\end{bmatrix}
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\begin{bmatrix}
0.18 & 0.36 & 0.18 & 0.90 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.53 & 0.80 & 0.27
\end{bmatrix}
$$

例子取自Duen Horng Chau <Georgia Tech CSE6242: Latent Semantic Indexing>

本讲小结

- 奇异值分解与低秩矩阵近似
- 奇异值分解在主成分分析、矩阵补全、潜在语义分析中的应用

计算机学院
余皓然

主要参考资料

Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox> Lecture Notes

Cameron Musco <COMPSCI 514 - Algorithms for Data Science> Slides

Dan Jackson <The Netflix Prize: How a $1 Million Contest Changed Binge-Watching Forever> Article

Alex Thomo <SENG 474: Data Mining> Lecture Notes and Slides

Duen Horng Chau <Georgia Tech CSE6242: Latent Semantic Indexing> Video

ccjou <线代启示录： SVD于矩阵近似的应用> Webpage

计算机学院
余皓然
2024/5/23

大语言模型的低秩适应

LoRA

针对特定应用，大语言模型的微调（fine-tuning）消耗大量算力和训练时间

- GPT-4大约有1.8万亿个参数、100+层

$$\max_{\Phi} \log\left(p_{\Phi}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$

$$\Leftrightarrow \max_{\Delta\Phi} \log\left(p_{\Phi_0+\Delta\Phi}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$

ΔΦ与Φ有相同的维数

LoRA

针对特定应用，大语言模型的微调（fine-tuning）消耗大量算力和训练时间

- GPT-4大约有1.8万亿个参数、100+层

$$\max_{\Phi} \log\left(p_{\Phi}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$
$$\Leftrightarrow \max_{\Delta\Phi} \log\left(p_{\Phi_0+\Delta\Phi}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$

ΔΦ与Φ有相同的维数

- 低秩适应（Low-Rank Adaptation）：核心思想是利用矩阵低秩的特点、用带有少量训练参数的ΔΦ(Θ)近似ΔΦ，即考虑如下近似问题

$$\max_{\Theta} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$

相比Φ，Θ中包含参数个数可降为0.01%　类似思想在变分推断等方法中也有应用

LoRA

Lora: Low-rank adaptation of large language models

EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen

arXiv preprint arXiv:2106.09685, 2021 • arxiv.org

ICLR'22

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example -- deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable

SHOW MORE

☆ Save 99 Cite Cited by 3863 Related articles All 8 versions »

LoRA

Lora: Low-rank adaptation of large language models

EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen

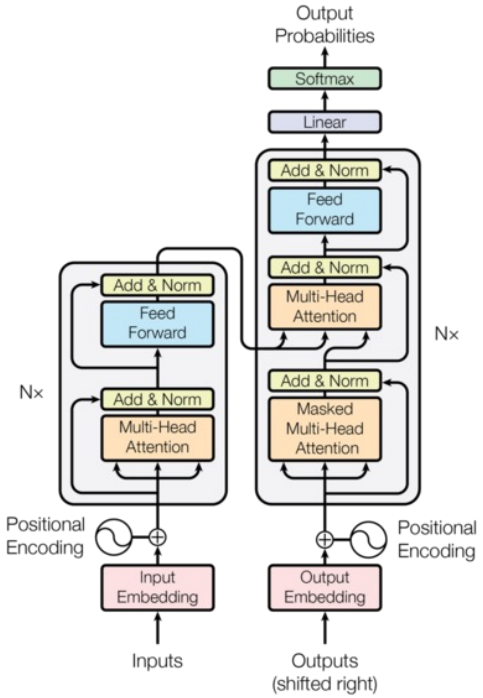
arXiv preprint arXiv:2106.09685, 2021 • arxiv.org

ICLR'22

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example -- deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable

SHOW MORE

☆ Save 99 Cite Cited by 3863 Related articles All 8 versions »



大语言模型的基本单元Transformer

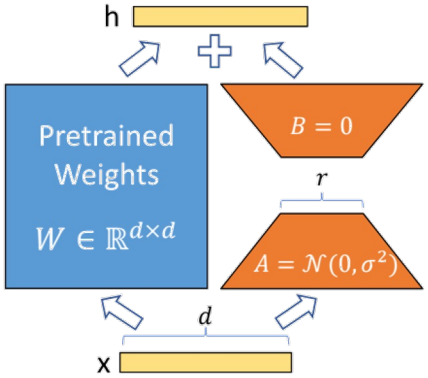


Figure 1: Our reparametrization. We only train A and B.

考虑网络中的一层，输入为$\boldsymbol{x} \in \mathbb{R}^k$、输出为$\boldsymbol{h} \in \mathbb{R}^d$

LoRA

Lora: Low-rank adaptation of large language models

EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen

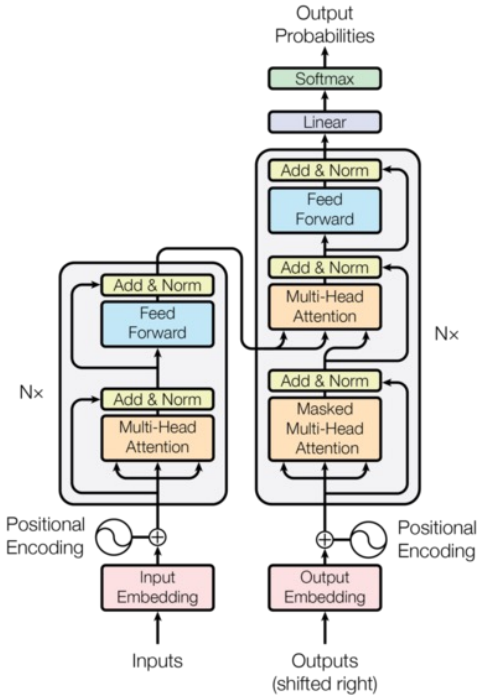
ICLR'22

arXiv preprint arXiv:2106.09685, 2021 · arxiv.org

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example -- deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable

SHOW MORE

☆ Save 99 Cite Cited by 3863 Related articles All 8 versions »



大语言模型的基本单元Transformer

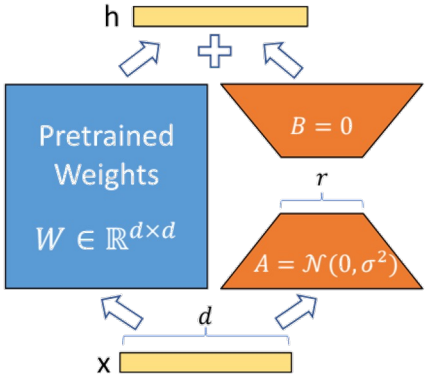


Figure 1: Our reparametrization. We only train A and B.

考虑网络中的一层，输入为$\boldsymbol{x} \in \mathbb{R}^k$、输出为$\boldsymbol{h} \in \mathbb{R}^d$

$$\boldsymbol{h} = \mathbf{W}\boldsymbol{x}$$

其中$\mathbf{W} \in \mathbb{R}^{d\times k}$

LoRA

Lora: Low-rank adaptation of large language models

EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen

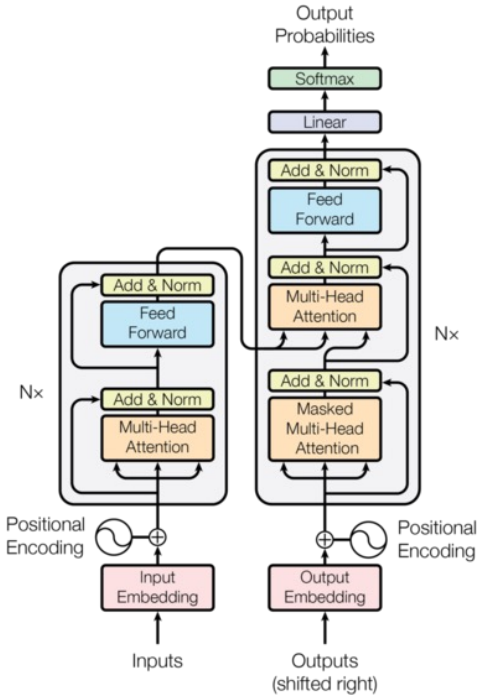
arXiv preprint arXiv:2106.09685, 2021 · arxiv.org

ICLR'22

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example -- deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable

SHOW MORE

☆ Save 99 Cite Cited by 3863 Related articles All 8 versions »



大语言模型的基本单元Transformer

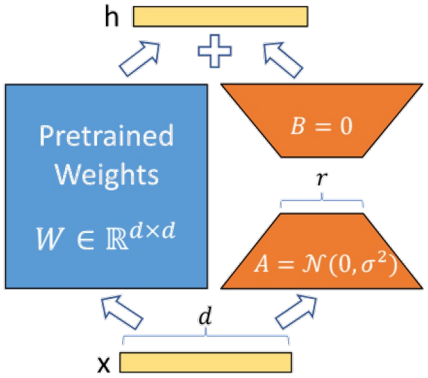


Figure 1: Our reparametrization. We only train A and B.

考虑网络中的一层，输入为$\boldsymbol{x} \in \mathbb{R}^k$、输出为$\boldsymbol{h} \in \mathbb{R}^d$

$$\boldsymbol{h} = \mathbf{W}\boldsymbol{x} = (\mathbf{W}_0 + \Delta\mathbf{W})\boldsymbol{x}$$

其中$\mathbf{W}, \mathbf{W}_0, \Delta\mathbf{W} \in \mathbb{R}^{d\times k}$

LoRA

Lora: Low-rank adaptation of large language models

EJ Hu, Y Shen, P Wallis, Z Allen-Zhu, Y Li, S Wang, L Wang, W Chen

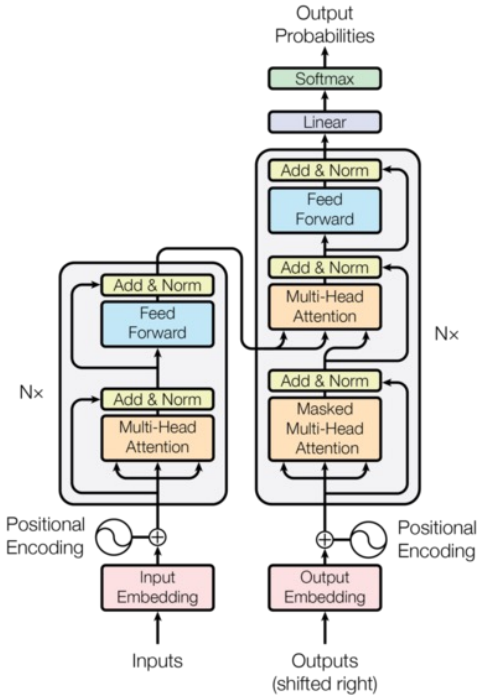
arXiv preprint arXiv:2106.09685, 2021 · arxiv.org

ICLR'22

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example -- deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable

SHOW MORE

☆ Save 99 Cite Cited by 3863 Related articles All 8 versions »



大语言模型的基本单元Transformer

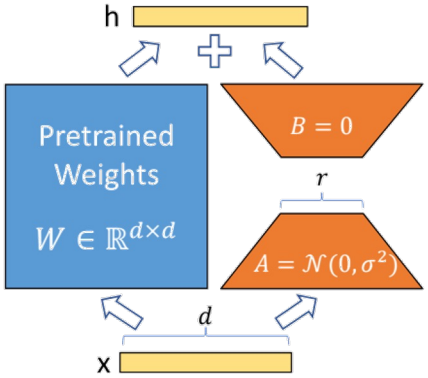


Figure 1: Our reparametrization. We only train A and B.

考虑网络中的一层，输入为$\boldsymbol{x} \in \mathbb{R}^k$、输出为$\boldsymbol{h} \in \mathbb{R}^d$

$$\boldsymbol{h} = \mathbf{W}\boldsymbol{x} = (\mathbf{W}_0 + \Delta\mathbf{W})\boldsymbol{x} = (\mathbf{W}_0 + \mathbf{BA})\boldsymbol{x}$$

其中$\mathbf{W}, \mathbf{W}_0, \Delta\mathbf{W} \in \mathbb{R}^{d\times k}, \mathbf{B} \in \mathbb{R}^{d\times r}, \mathbf{A} \in \mathbb{R}^{r\times k} (r \ll \min\{d,k\})$

LoRA

$$\max_{\Theta} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}\left(\{\boldsymbol{y}_i\}_{i\in\mathcal{D}}\middle|\{\boldsymbol{x}_i\}_{i\in\mathcal{D}}\right)\right)$$

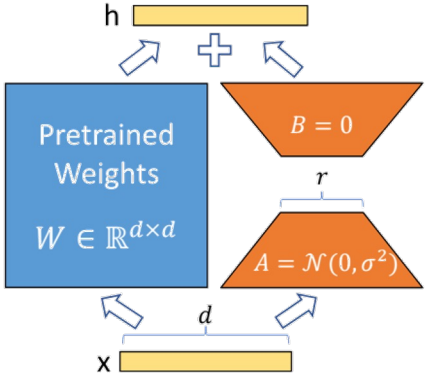


Figure 1: Our reparametrization. We only train A and B.

考虑网络中的一层，输入为$\boldsymbol{x} \in \mathbb{R}^k$、输出为$\boldsymbol{h} \in \mathbb{R}^d$

$$\boldsymbol{h} = \mathbf{W}\boldsymbol{x} = (\mathbf{W}_0 + \Delta\mathbf{W})\boldsymbol{x} = (\mathbf{W}_0 + \mathbf{BA})\boldsymbol{x}$$

其中$\mathbf{W}, \mathbf{W}_0, \Delta\mathbf{W} \in \mathbb{R}^{d\times k}, \mathbf{B} \in \mathbb{R}^{d\times r}, \mathbf{A} \in \mathbb{R}^{r\times k} (r \ll \min\{d,k\})$

实际需要训练的参数数目变为：

$|\Theta| = 2 \times \hat{L}_{LoRA} \times d_{model} \times r$, where $\hat{L}_{LoRA}$ is the number of weight matrices we apply LoRA to.

谢谢！

计算机学院
余皓然
2024/5/23