隐性语义索引
Latent Semantic Indexing

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

线性代数基础

- 令 **C** 为一个 $M \times N$ 的词项-文档矩阵，其中的每个元素都是非负实数。
- 矩阵的秩(rank)是线性无关的行(或列)的数目，因此有 $rank(\mathbf{C}) \leq \min\{M, N\}$。
- 一个非对角线上元素均为零的 $r \times r$ 方阵被称为对角阵(diagonal matrix)，其秩等于其对角线上非零元素的个数。
- 如果上述对角阵上的r个元素都是1，则称之为 r维单位矩阵(identity matrix)，记为$\mathbf{I}_r$。

$$\begin{pmatrix} 1 & 6 & -4 & -1 & 4 \\ 0 & -4 & 3 & 1 & -1 \\ 0 & 0 & 0 & 4 & -8 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

$$\begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

- 对于 $M \times M$ 的方阵 $\mathbf{C}$ 及非零向量 $\vec{x}$，有

$$\mathbf{C}\vec{x} = \lambda\vec{x}$$

- 满足上式的 λ 被称为矩阵 $\mathbf{C}$ 的特征值(eigenvalues)。
- 对于特征值 λ，满足等式 M 维非零向量 $\vec{x}$ 称为其右特征向量(right eigenvector)。
- 对应最大特征值的特征向量被称为主特征向量(principal eigenvector)。
- 同样，矩阵 $\mathbf{C}$ 的左特征向量(left eigenvectors)是满足下列等式的 M 维 向量 $\vec{y}$：

$$\vec{y}^T\mathbf{C} = \lambda\vec{y}^T$$

- 特征方程(characteristic equation)

$$(\mathbf{C} - \lambda \mathbf{I}_M)\vec{x} = 0$$

  - 可以通过求解这个方程来得到矩阵的特征值

矩阵分解(matrix decomposition)

- 将方阵分解成多个矩阵因子乘积的方法，并且这几个矩阵因子都可以从方阵的特征向量导出

矩阵对角化定理

- 令 **S** 为 $M \times M$ 的实方阵，并且它有 M 个线性无关的特征向量，那么存在一个特征分解：

$$\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^{-1}$$

  - 其中**U** 的每一列都是**S**的特征向量
  - Λ 是按照特征值从大到小排列的对角阵

$$\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_M \end{pmatrix}, \quad \lambda_i \geqslant \lambda_{i+1}$$

  - 如果特征值都不相同，那么该分解是唯一的

对称对角化定理

- 假定 **S** 是一个 $M \times M$ 的实对称方阵，并且它有 M 个线性无关的特征向量，那么存在如下一个对称对角化分解：

$$\mathbf{S} = \mathbf{Q}\Lambda\mathbf{Q}^{\mathrm{T}}$$

  - 其中，**Q**的每一列都是**S**的互相正交且归一化(单位长度)的特征向量，
  - Λ 是对角矩阵，其每个对角线上的值都对应**S**的一个特征值。
  - 另外，由于**Q**是实矩阵，所以有$\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{-1}$

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

词项-文档矩阵及SVD

- 迄今为止介绍的分解都是基于方阵，然而，我们感兴趣的是 $M \times N$ 的词项-文档矩阵 **C**，如果排除极端罕见的情况，那么有 $M \neq N$
- 另外，**C** 基本上也不可能是对称矩阵。
- 因此，先给出对称对角化分解的一个被称为 SVD 的扩展形式，然后将它用于构建 **C** 的近似矩阵

- 给定矩阵**C**，
  - **U**是一个$M \times M$的矩阵，其每一列是矩阵$\mathbf{C}\mathbf{C}^{\mathrm{T}}$的正交特征向量，
  - 而$N \times N$矩阵**V**的每一列都是矩阵$\mathbf{C}^{\mathrm{T}}\mathbf{C}$的正交特征向量。
  - $\mathbf{C}^{\mathrm{T}}$是**C**的转置矩阵。
- 定理：令r是$M \times N$矩阵**C**的秩，那么**C**存在如下形式的SVD：

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$$

  - $\mathbf{C}\mathbf{C}^{\mathrm{T}}$的特征值$\lambda_1, \lambda_2, \ldots, \lambda_r$等于 $\mathbf{C}^{\mathrm{T}}\mathbf{C}$的特征值；
  - 对于 $1 \le i \le r$，令 $\sigma_i = \sqrt{\lambda_i}$ ，并且 $\lambda_i \ge \lambda_{i+1}$。 $M \times N$的矩阵$\mathbf{\Sigma}$满足$\mathbf{\Sigma}_{ii}=\sigma_i$，其中 $1 \le i \le r$，而$\mathbf{\Sigma}$ 中其他元素均为 0。
  - 其中，σ_i就是矩阵 **C** 的奇异值(singular value)

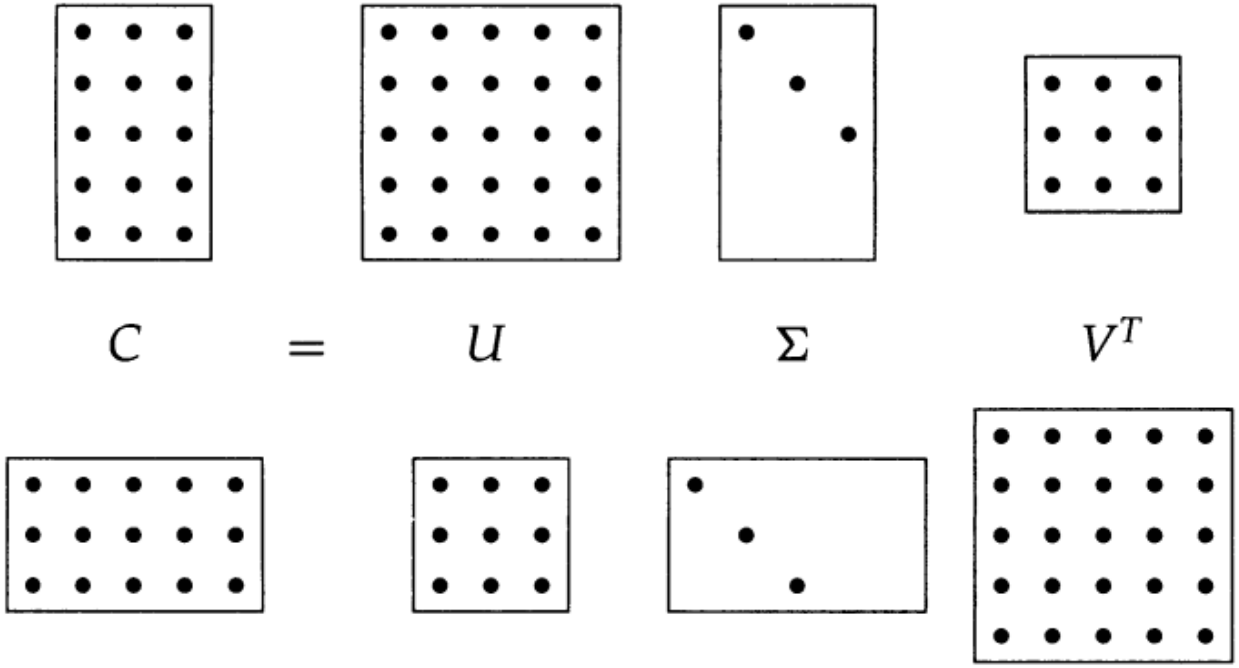

图 18-1　SVD 的示意图。该示意图给出了两种情况：上图中，$M \times N$ 的矩阵 C 满足 $M > N$。而下图中 $M < N$

$\mathbf{CC^T = U\Sigma V^T V\Sigma U^T = U\Sigma\Sigma^T U^T}$

- 左边$\mathbf{CC^T}$是一个实对称方阵
- 右边$\mathbf{U\Sigma\Sigma^T U^T}$正好是对称对角化分解形式
- $\mathbf{CC^T}$实际上是一个方阵，其每行和每列都对应 *M* 个词项中的一个。
  - 矩阵中的第 *i* 行、第 *j* 列的元素实际上是第 *i* 个词项与第 *j* 个词项基于文档共现次数的一个重合度计算指标。
  - 其精确的数学含义依赖于构建**C**所使用的词项权重方法
    - 假定**C**是词项-文档布尔矩阵，那么$\mathbf{CC^T}$的第 *i* 行、第 *j* 列的元素是词项 *i* 和词项 *j* 共现的文档数目

- 当记录 SVD分解的数值结果时，由于其他部分都是零，常规做法是将**Σ**表示成一个$r \times r$的对角方阵，所有奇异值排列在对角线上。同样，对应于**Σ**中被去掉的行，**U**中的最右$M-r$列也被去掉。对应于**Σ**中被去掉的列，**V**中的最右$N-r$列也被去掉。这种SVD的书写形式有时被称为简化的SVD(reduced SVD)或截断的SVD(truncated SVD)

例 18-3 这里给出一个秩为 2 的 4×2 矩阵的 SVD 例子，奇异值$\Sigma_{11}=2.236$，$\Sigma_{22}=1$。

$$C=\begin{pmatrix}1 & -1\\ 0 & 1\\ 1 & 0\\ -1 & 1\end{pmatrix}=\begin{pmatrix}-0.632 & 0.000\\ 0.316 & -0.707\\ -0.316 & -0.707\\ 0.632 & 0.000\end{pmatrix}\begin{pmatrix}2.236 & 0.000\\ 0.000 & 1.000\end{pmatrix}\begin{pmatrix}-0.707 & 0.707\\ -0.707 & -0.707\end{pmatrix}。 \quad (18\text{-}11)$$

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

低秩逼近

- 给定 $M \times N$ 的矩阵$\mathbf{C}$及正整数 k，寻找一个秩不高于 k 的 $M \times N$ 的矩阵$\mathbf{C}_k$，使得两个矩阵的差 $\mathbf{X} = \mathbf{C} - \mathbf{C}_k$的F–范数(Frobenius Norm，弗罗宾尼其范数)最小，即下式最小

$$\| X \|_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} X_{ij}^{\ 2}}$$

  - X的F–范数度量了$\mathbf{C}_k$和$\mathbf{C}$之间的差异程度。
  - 目标是找到一个矩阵$\mathbf{C}_k$，会使得这种差异极小化，同时又要限制$\mathbf{C}_k$的秩不高于k。
  - 如果r是$\mathbf{C}$的秩，那么很显然$\mathbf{C}_r$=$\mathbf{C}$，此时矩阵差值的F范数为 0。
  - 当k比r小得多时，称$\mathbf{C}_k$为低秩逼近(low-rank approximation)矩阵

- SVD可以用于解决矩阵低秩逼近问题，将其应用到词项-文档矩阵的逼近问题上来。要进行三步操作：
  - 给定$\mathbf{C}$，构造SVD分解，因此 $\mathbf{C} = \mathbf{U\Sigma V}^{\mathrm{T}}$；
  - 把Σ中对角线上r-k个最小奇异值置为0，从而得到Σ_k；
  - 计算 $\mathbf{C}_k = \mathbf{U\Sigma}_k\mathbf{V}^{\mathrm{T}}$作为 $\mathbf{C}$ 的逼近。
- 由于$\mathbf{\Sigma}_k$最多包含k个非零元素，所以$\mathbf{C}_k$的秩不高于k
- 小特征值对于矩阵乘法的影响也小。因此，将这些小特征值替换成 0 将不会对最后的乘积有实质性影响，也就是说该乘积接近 $\mathbf{C}$。

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

回顾一下词项文档矩阵

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
anthony	5.25	3.18	0.0	0.0	0.0	0.35
brutus	1.21	6.10	0.0	1.0	0.0	0.0
caesar	8.59	2.54	0.0	1.51	0.25	0.0
calpurnia	0.0	1.54	0.0	0.0	0.0	0.0
cleopatra	2.85	0.0	0.0	0.0	0.0	0.0
mercy	1.51	0.0	1.90	0.12	5.25	0.88
worser	1.37	0.0	0.11	4.15	0.25	1.95
. . .						

该矩阵是计算文档和查询的相似度的基础，接下来介绍能否通过对该矩阵进行转换来获得文档和查询之间的一个更好的相似度计算方法？

隐性语义索引LSI简介

- 将词项-文档矩阵转换成多个矩阵的乘积
- 这里使用的是一个特定的分解方法奇异值分解(Singular value decomposition, SVD)
- $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (其中**C**是词项-文档矩阵)
- 利用SVD分解的结果来构造一个新的、改进的词项-文档矩阵**C**’
- 通过**C**’可以得到一个更好的相似度计算方法(相对**C**而言)
- 为了这种目的使用SVD被称为隐性语义索引(Latent Semantic Indexing, LSI)

例子 $C = U\Sigma V^T$，矩阵C

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

- 词项-文档矩阵(布尔)

例子$C = U\Sigma V^T$，矩阵U

	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

- 每个词项对应一行，每个$min(M, N)$对应一列，M为词项数目，N是文档数目
- 这是一个正交矩阵
  - 列向量都是单位向量
  - 任意两个列向量之间都是正交的。可以想象这些列向量分布代表不同的“语义”维度，比如政治、体育、经济等主题。
  - 矩阵元素u_{ij}给出的是词项i和第j个“语义”维度之间的关系强弱程度

例子$C = U\Sigma V^T$，矩阵Σ

	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

- 是一个$min(M,N) \times min(M,N)$的对角方阵
  - 对角线上是矩阵**C**的奇异值
  - 奇异值的大小度量的是对应“语义”维度的重要性
  - 可以通过忽略较小的值来忽略对应的“语义”维度

例子 $C = U\Sigma V^T$，矩阵V^T

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

- 每个词项对应一列，每个$min(M,N)$对应一行
- 这也是一个正交矩阵
  - 每个行向量都是单位向量
  - 任意两个列向量之间都是正交的
  - 可以想象每个行向量代表一个“语义”维度
  - 矩阵元素v_{ij}给出的是文档i和第j个“语义”维度之间的关系强弱程度

例子$C = U\Sigma V^T$，所有4个矩阵

C

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

U

	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

Σ

	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

V^T

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

LSI小结

- 词项-文档矩阵可以分解成3个矩阵的乘积
- 词项矩阵**U**-每个词项对应其中的一个行向量
- 文档矩阵$\mathbf{V}^T$-每篇文档对应其中的一个列向量
- 奇异值矩阵**Σ**-对角方阵，对角线上的奇异值代表的是每个“语义”维度的重要性

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

为什么在LSI中使用SVD分解

- 最关键的性质：每个奇异值对应的是每个“语义”维度的权重
- 将不太重要的权重置为0，可以保留重要的信息，去掉一些信息“枝节”
- 这些“枝节”可能是
  - 噪音-这种情况下，简化的LSI噪音更少，是一种更好的表示方法
  - 枝节信息可能会使本来应该相似的对象不相似，同样简化的LSI由于其能更好的表达相似度，因而是一种更优的表示方式
- “细节越少越好”的一个类比
  - 鲜红色花朵的图像
  - 红黑花朵的图像
  - 如果忽略颜色，将更容易看到两者的相似性

将空间维度将为2

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

实际上，只需将矩阵Σ中相应的维度置为0即可。此时，相当于矩阵U和V^T的相应维度被忽略，然后计算

$C_2 = U\Sigma_2 V^T$

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

为什么新的低维空间更好？

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

- 在原始空间C中，d_2和d_3的相似度是0
- 在新的空间C_2，d_2和d_3的相似度为$0.52 * 0.28 + 0.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + -0.39 * -0.08 \approx 0.52$

为什么新的低维空间更好？

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

- Boat和ship语义相似。低维空间能反映出这一点。

目录

- 矩阵分解
- 词项—文档矩阵及 SVD
- 低秩逼近
- 隐性语义索引
- 空间降维处理
- LSI在IR中的应用

LSI在IR中使用的原因

- LSI能够发现文档语义上的关联
- 但是在原始向量空间中这些文档相似度不大(因为它们使用不同的词语)
- 通过LSI将它们映射到新的低维向量空间中
- 在新的空间下，两者相似度较高
- 因此，LSI能解决一词多义和语义关联问题
- 在原始向量空间下，同义词对文档相似度没有任何贡献
- LSI所期望的效果：同义词对文档相似度贡献很大

LSI是如何解决一词多义和语义关联问题的

- 降维迫使忽略大量“细节”
- 将原始空间下不同的词映射到低维空间的同一维中
- 将同义词映射到同一维的“开销”远小于无关词的聚集
- SVD选择开销最小的映射方法
- 因此，SVD会将同义词映射到同一维
- 但是，它同时能避免将无关词映射到同一维

LSI与其它方法的比较

- 如果查询和文档没有公共词项时，前面介绍的相关反馈和查询扩展可以用于提高IR的召回率
- LSI会提高召回率但是损害正确率
- 因此，它和相关反馈查询扩展解决的是同一问题
- 同样，它们的缺点也一致

LSI的实现

- 对词项-文档矩阵进行SVD分解
- 计算在新的低维空间下的文档表示
- 将查询q映射到LSI低维空间中

$$\vec{q}_k = \Sigma_k^{-1}\mathbf{U}_k^{\mathrm{T}}\vec{q}^{\mathrm{T}}$$

  - 上式来自 $\mathbf{C}_2 = \mathbf{U}\boldsymbol{\Sigma}_2\mathbf{V}^{\mathrm{T}} \Rightarrow \boldsymbol{\Sigma}_2^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{C} = \mathbf{V}_2^{\mathrm{T}}$

- 计算q_2和$\mathbf{V}_2$中的所有文档表示的相似度
- 像以往一样按照相似度高低输出文档结果

最优性

- SVD在下面的意义上说是最优的
  - 保留k个最大的奇异值并将其它奇异值置为0，这种做法得到原始矩阵$\mathbf{C}$的最佳逼近
  - 最优性：不存在其它同秩的矩阵更加逼近$\mathbf{C}$