信息检索与数据挖掘

矩阵分解在信息检索中的应用

课程内容

- 第1章 绪论
- 第2章 布尔检索及倒排索引
- 第3章 词项词典和倒排记录表
- 第4章 索引构建和索引压缩
- 第5章 向量模型及检索系统
- 第6章 检索的评价
- 第7章 相关反馈和查询扩展
- 第8章 概率模型
- 第9章 基于语言建模的检索模型
- 第10章 文本分类
- 第11章 文本聚类
- 第12章 Web搜索
- 第13章 多媒体信息检索
- 第14章 其他应用简介

Information Retrieval(IR)：从大规模非结构化数据（通常是文本）的集合（通常保存在计算机上）中找出满足用户信息需求的资料（通常是文档）的过程

数据挖掘（Data Mining）从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

词项-文档矩阵C→CC^T

- C：$M \times N$ 的词项-文档矩阵
- CC^T 的物理意义？

习题 **18-4** 令

$$C = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}。 \quad (18\text{-}12)$$

为某个文档集上的词项-文档出现矩阵，计算词项的共现矩阵 CC^T。当 C 是一个词项-文档出现矩阵时，CC^T对角线上的元素是什么？

CC^T 方阵，其每行和每列都对应M个词项中的一个。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。

C_{5*6}

	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

C^T_{6*5}

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

词项-文档矩阵C→C^TC

习题 18-6 假定 C 是词项–文档出现矩阵，那么 C^TC 的元素的含义是什么？

- C：$M \times N$ 的词项-文档矩阵
- C^TC的物理意义？

C^T_{6*5}

	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

C^TC是方阵，其每行和每列都对应N个文档中的一个。CC^T 的矩阵中的第i 行、第j 列的元素实际上是第i 个文档与第j 个文档含有相同词项的数目。

C_{5*6}

	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-文档计数(tf)矩阵C→CC^T、C^TC
词项-文档权重(tf-idf)矩阵C→ CC^T、 C^TC

- 物理意义？
  - A_{ii}是词项i在不同文档中出现次数的平方和
  - A_{ij}是词项i和词项j共现时tf_i*tf_j的累计

习题 **18-7** 令

$$C=\begin{pmatrix}0 & 2 & 1\\ 0 & 3 & 0\\ 2 & 1 & 0\end{pmatrix}。 \qquad (18\text{-}14)$$

上式为一个词项–文档矩阵，其中每个元素都是词项频率，因此词项 1 在文档 2 中出现 2 次，而在文档 3 中出现 1 次。计算 CC^T，并找出两个词项的最高词频都出现在同一文档时所对应的元素。

CC^T各元素体现了词项和词项之间的关联程度
C^TC各元素体现了文档和文档之间的关联程度

小结：词项-文档矩阵

- C：$M \times N$ 的词项-文档矩阵
  - C的每一列即为向量空间模型中的一个向量
  - 文档和查询均表示为向量，相关度为向量的“距离”
- $A=C^TC$
  - A_{ij}是词项i 和词项j 共现的文档数目
- $A=CC^T$
  - A_{ij}是第i 个文档与第j 个文档含有相同词项的数目
- 词项-文档计数(tf)矩阵C→CC^T、C^TC
- 词项-文档权重(tf-idf)矩阵C→ CC^T、 C^TC

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

线性代数基础
特征值与特征向量

令C为一个$M \times N$的矩阵，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$rank(C) \leq \min\{M,N\}$。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），它的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I_r。

对于M×M的方阵C及非零向量$\vec{x}$，满足$C\vec{x} = \lambda\vec{x}$的λ被称为矩阵C的特征值（eigenvalues）。C的非零特征值的个数最多是rank(C)。对于特征值λ，满足$C\vec{x} = \lambda\vec{x}$的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C的左特征向量（left eigenvectors）是满足$\vec{y}^T C = \lambda\vec{y}^T$式的M维向量$\vec{y}$。

线性代数基础
求解特征值

等式$C\vec{x} = \lambda\vec{x}$可以改写成 $(C - \lambda I_M)\vec{x} = 0$，这个等式称为特征方程（characteristic equation），可以通过求解这个方程来得到矩阵的特征值。因此，C的特征值也就是方程 $|(C - \lambda I_M)| = 0$ 的解，其中|S|表示的是方阵S 的行列式（determinant）。

$|(C - \lambda I_M)| = 0$ 是一个以λ为变量的M阶多项式方程，因此它最多有M个根，这些根也就是矩阵C 的特征值。即使C 中所有元素都是实数，那么这些特征值通常也有可能是复数。

对于对称（symmetric）矩阵S，不同特征值所对应的特征向量之间是正交的（orthogonal）。另外，如果S 是实对称矩阵，那么所有特征值也都是实数。

线性代数基础
向量可表征为特征向量的线性组合

考虑矩阵$S = \begin{pmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 1 \end{pmatrix}$，很明显，矩阵的秩是3，并且具有3 个非零特征值 λ_1=30、λ_2=20 及λ_3=1，它们对应的特征向量分别是：$\overrightarrow{x_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \overrightarrow{x_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \overrightarrow{x_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$。

现考虑任意一个向量，如$\vec{v} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$，总是可以将$\vec{v}$表示成S 的三个特征向量的线性组合，如本例：$\vec{v} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} = 2\overrightarrow{x_1} + 4\overrightarrow{x_2} + 6\overrightarrow{x_3}$

。

线性代数基础
向量可表征为特征向量的线性组合

考虑矩阵$S=\begin{pmatrix}2 & 1\\ 1 & 2\end{pmatrix}$，将上述矩阵代入特征方程$|S-\lambda I|=0$可以得到二次方程$(2-\lambda)^2-1=0$，对这个方程求解可以得到2个特征值$\lambda_1=3$、$\lambda_2=1$，它们对应的特征向量$\overrightarrow{x_1}=\begin{pmatrix}1\\ 1\end{pmatrix}$和$\overrightarrow{x_2}\begin{pmatrix}1\\ -1\end{pmatrix}$是正交的。现考虑任意一个向量，如$\vec{v}=\begin{pmatrix}2\\ 3\end{pmatrix}$，总是可以将$\vec{v}$表示成S 的1个特征向量的线性组合，如本例：$\vec{v}=\begin{pmatrix}2\\ 3\end{pmatrix}=2.5\overrightarrow{x_1}-0.5\overrightarrow{x_2}$。

小结：线性代数基础

- 矩阵$C_{M\times N}$，其中的每个元素都是非负实数
- $rank(C)\leq \min\{M,N\}$
- 方阵$C_{M\times M}$
- 左特征向量：满足$\vec{y}^T C = \lambda\vec{y}^T$ 式的M维向量$\vec{y}$。
- 右特征向量：满足$C\vec{x} = \lambda\vec{x}$的M维非零向量$\vec{x}$ 。

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

矩阵分解
方阵的分解

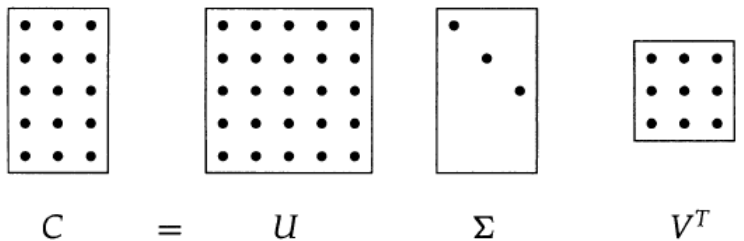
- 矩阵分解（matrix decomposition）
  - 将矩阵分解成多个矩阵因子乘积。
- 实方阵的基本因子分解方法
  - 定理 **18-1**（矩阵对角化定理） 令S 为$M\times M$的实方阵，并且它有M个线性无关的特征向量，那么存在一个特征分解：$S=U\Lambda U^{-1}$。
  - 其中，U 的每一列都是S 的特征向量，Λ 是按照特征值从大到小排列的对角阵。
- 实对称方阵的分解方法
  - 定理**18-2**（对称对角化定理） 假定S 是一个$M\times M$的实对称方阵，并且它有M个线性无关的特征向量，那么存在如下一个对称对角化分解：$S=Q\Lambda Q^T$。
  - 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：$Q^{-1}=Q^T$。

矩阵分解

$M \times N$ 矩阵C分解，$M \neq N$

- 给定$M \times N$ 矩阵C，U 是一个$M \times M$ 的矩阵，其每一列是矩阵CC^T 的正交特征向量，而$N \times N$ 矩阵V的每一列都是矩阵C^TC 的正交特征向量。这里C^T是C 的转置矩阵。
- 定理**18-3** 令r 是$M \times N$ 矩阵C 的秩，那么C 存在如下形式的奇异值分解（SVD）：$\boldsymbol{C = U\Sigma V^T}$。其中
  - 1．CC^T 的特征值 $\lambda_1, \lambda_2, \ldots, \lambda_r$ 等于C^TC 的特征值；
  - 2．对于$1 \leq i \leq r$，令$\sigma = \sqrt{\lambda_i}$，并且$\lambda_i \geq \lambda_{i+1}$。$M \times N$的矩阵$\Sigma$ 满足$\Sigma\, ii = \sigma_i$，其中$1 \leq i \leq r$，而Σ 中其他元素均为0。

σ_i 就是矩阵C 的奇异值（singular value）



低秩逼近

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

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

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

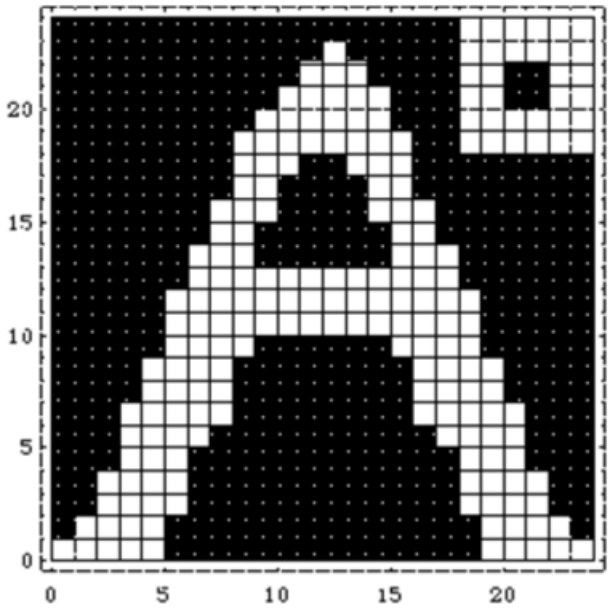
SVD 用于矩阵的低秩逼近

- 进行如下三步操作：
- (1) 给定C，构造SVD 分解，因此$C = U\Sigma V^T$；
- (2) 把Σ对角线上r-k 个最小奇异值置为0，得到Σ_k；
- (3) 计算$C_k = U\Sigma_k V^T$ 作为C 的逼近。

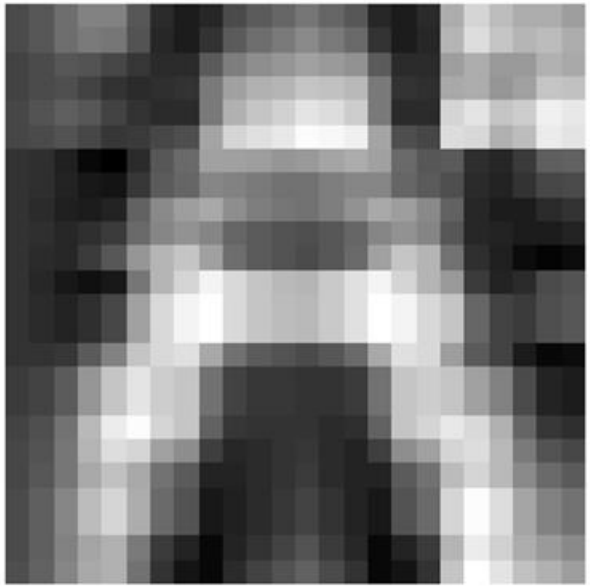
- 由于 Σ_k 最多包含k 个非零元素，所以C_k 的秩不高于k。将这些小特征值替换成0 将不会对最后的乘积有实质性影响，也就是说该乘积接近C。

Eckart 及Young 给出的定理将会告诉我们，上述过程产生了一个秩为k 的矩阵，它的F–范数误差最小。

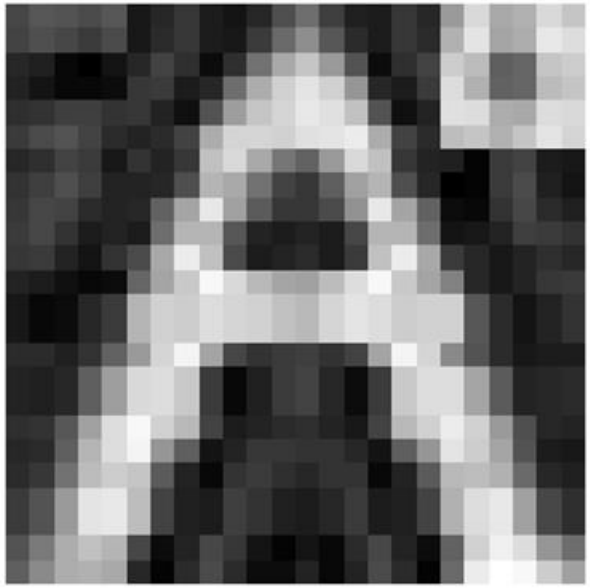
SVD用于图像压缩



A 24*24 image



Rank 3 approximation



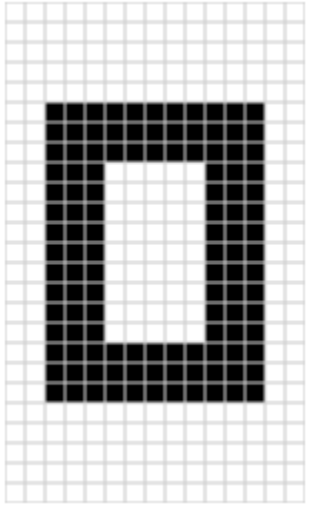
Rank 5 approximation

$$A = \left[\begin{array}{ccc|ccc} u_1 & \cdots & u_k & u_{k+1} & \cdots & u_m \end{array}\right] \left[\begin{array}{ccc|c} \sigma_1 & & & 0 \\ & \ddots & & \\ & & \sigma_k & \\ \hline 0 & & & 0 \end{array}\right] \left[\begin{array}{c} v_1^T \\ \vdots \\ v_k^T \\ \hline v_{k+1}^T \\ \vdots \\ v_n^T \end{array}\right]$$

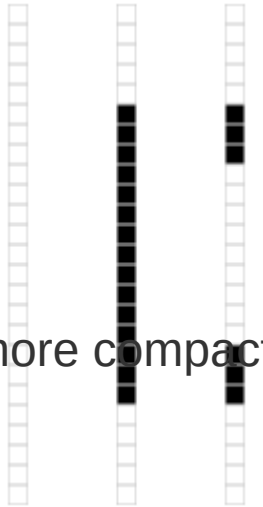
http://www.math.umn.edu/~lerman/math5467/svd.pdf

SVD用于图像压缩

an array of 15*25 black or white pixels



represent the data in a more compact form



375 entries in the matrix

$$M = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

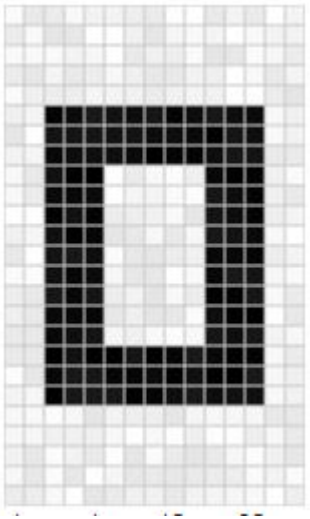
$\sigma_1 = 14.72$
$\sigma_2 = 5.22$
$\sigma_3 = 3.31$

$$M = \mathbf{u}_1\sigma_1\,\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\,\mathbf{v}_2^T + \mathbf{u}_3\sigma_3\,\mathbf{v}_3^T$$

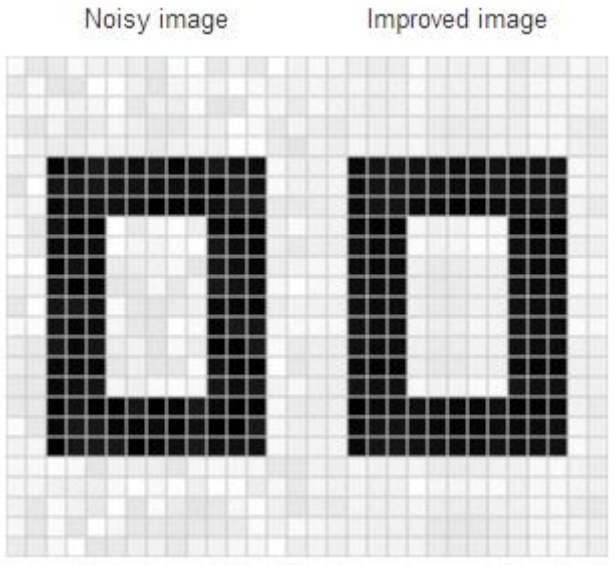
This means that we have three vectors $\mathbf{v}_i$, each of which has 15 entries, three vectors $\mathbf{u}_i$, each of which has 25 entries, and three singular values $\boldsymbol{\sigma}_i$. This implies that we may represent the matrix using only 123 numbers rather than the 375 that appear in the matrix. In this way, the singular value decomposition discovers the redundancy in the matrix and provides a format for eliminating it.

http://www.ams.org/samplings/feature-column/fcarc-svd

SVD用于图像去噪



$\sigma_1 = 14.15$
$\sigma_2 = 4.67$
$\sigma_3 = 3.00$
$\sigma_4 = 0.21$
$\sigma_5 = 0.19$
...
$\sigma_{15} = 0.05$

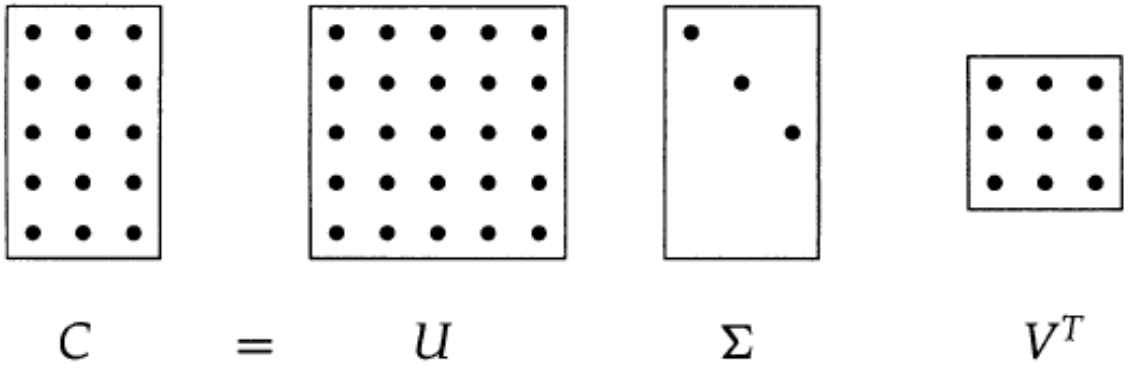


$$M = \mathbf{u}_1\sigma_1\,\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\,\mathbf{v}_2^T + \mathbf{u}_3\sigma_3\,\mathbf{v}_3^T$$

http://www.ams.org/samplings/feature-column/fcarc-svd

小结：矩阵分解与低秩逼近

$$C_{M\times N} = U_{M\times r}\Sigma_{r\times r}V^T_{r\times N}$$



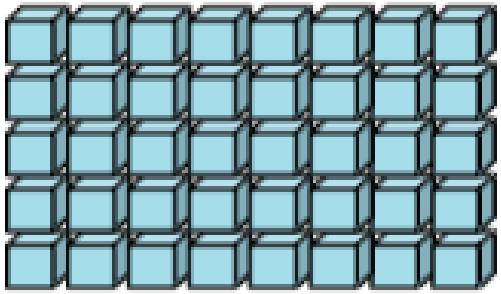
$$X = C - C_k$$

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

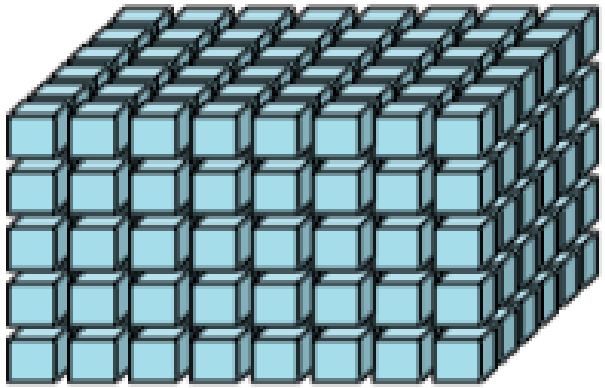
讨论：从矩阵到张量

matrix → tensor

一阶张量
（向量）

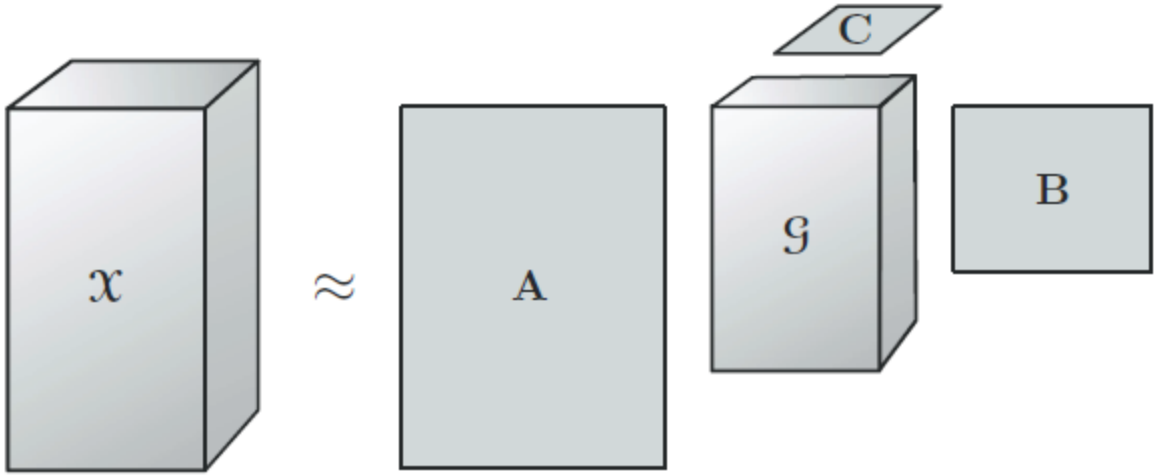


二阶张量
（矩阵）



三阶张量

讨论： Tucker decomposition



Tucker的1966年文章中第一次提到了Tucker分解。对于一个三阶张量, 由Tucker分解可以得到,三个因子矩阵和一个核张量 ,每个mode上的因子矩阵称为张量在每个mode上的基矩阵或者是主成分,因此Tucker 分解又称为高阶PCA, 高阶SVD等。

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

向量空间模型存在的问题

TABLE 1. Sample term by document matrix.[a]

	Access	Document	Retrieval	Information	Theory	Database	Indexing	Computer	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

[a]Query: "IDF in *computer-based information look-up*"

Term-Document矩阵，x代表该词项出现在对应的文档里，*表示该词项出现在查询(Query)中，当用户输入查询”IDF in computer-based information look up” 时，用户希望查找与信息检索中IDF（逆文档频率）相关的网页，按照精确词匹配的话，文档2和3分别包含查询中的两个词，因此应该被返回，而文档1不包含任何查询中的词，因此不会被返回。但我们仔细看看会发现，文档1中的access, retrieval, indexing, database这些词都是和查询相似度十分高的，其中retrieval和look up是同义词。从用户的角度看，文档1应该是相关文档，应该被返回。而文档2虽然包含查询中的一次词information，但文档2和IDF或信息检索无关，不是用户需要的文档，不应该被返回。从以上分析可以看出，在本次检索中，和查询相关的文档1并未返回给用户，而无查询无关的文档2却返回给了用户。这就是**同义词和多义词导致传统向量空间模型检索精确度的下降**。

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R.(1990).
Indexing By Latent Semantic Analysis. Journal of the American Society For Information Science, 41, 391-407. 10

引入LSI 的目的

LSI (Latent Semantic Indexing)

- 向量空间表示方法将查询和文档均表示成向量
  - 可以将查询和文档转换成同一空间下的向量，可以基于余弦相似度进行评分计算，能够对不同的词项赋予不同的权重
- 无法处理自然语言中的两个经典问题：一义多词（*synonymy*）和一词多义（*polysemy*）问题。
  - 一义多词指的是不同的词（比如car 和automobile）具有相同的含义。向量空间表示不能捕捉诸如car 和automobile 这类同义词之间的关系，而是将它们分别表示成独立的一维。因此，计算查询q（如car）和文档d（同时包含 car 和 automobile）的相似度$q \cdot d$ 时，就会**低估了用户所期望的相似度**。
  - 而一词多义指的是某个词项（如 charge）具有多个含义，因此在计算相似度$q \cdot d$ 时，就会**高估了用户所期望的相似度**。一个很自然的问题就是，能否利用词项的共现情况（比如，charge是和steed 还是electron 在某篇文档中共现），来获得词项的隐性语义关联从而减轻这些问题的影响？

C_k到C 的逼近性使得我们希望仍然可以保留原有的余弦相似度的相对大小：如果在原始空间中查询和文档相近，那么在新的k 维空间中它们仍然比较接近。

隐性语义索引 LSI (Latent Semantic Indexing) LSA(Latent Semantic Analysis)

- 利用SVD分解来找到词项-文档矩阵*C*的某个低秩逼近，在这个低秩逼近下能够为文档集中的每篇文档产生一个新的表示。
- 同样，查询也可以映射到这个低秩表示空间，从而可以基于新的表示来进行查询和文档的相似度计算。这个过程被称为LSI。

将词项-文档矩阵的低秩逼近与IR进行关联的工作来自Deerwester 等人(1990)，后来的相关结果的综述参见Berry 等人(1995)。

Indexing by latent semantic analysis
S Deerwester, ST Dumais, GW Furnas... - Journal of the ..., 1990 - search.proquest.com
Indexing by Latent Semantic Analysis Scott Deerwester Center for Information and Language Studies, University of Chicago, Chicago, IL 60637 Susan T. Dumais*, George W. Furnas, and Thomas K. Landauer Bell Communications Research, 445 South St., ...
被引用次数：10732 相关文章 所有 106 个版本 引用 保存

Probabilistic latent semantic indexing
T Hofmann - Proceedings of the 22nd annual international ACM ..., 1999 - dl.acm.org
... Latent Semantic Analysis LSA 1 is an approach to automatic indexing and information retrieval that attempts to overcome these problems by mapping documents as well as terms to a representation in the so called latent seman- tic space. ...
被引用次数：3897 相关文章 所有 30 个版本 引用 保存

隐性语义索引
LSI (Latent Semantic Indexing)

- 即使对一个中等规模的文档集来说，词项-文档矩阵C 也可能有成千上万个行和列，它的秩数目大概也是这个数量级。在LSI 中，我们使用SVD 分解来构造C 的一个低秩逼近C_k，其中k远小于矩阵C 原始的秩。一些研究工作当中，实验时k 的取值往往在几百以内。
- 这样，我们就可以将词项-文档矩阵中每行和每列（分别对应每个词项和每篇文档）映射到一个k 维空间，CC^T 和C^TC 的k 个主特征向量（对应k 个最大的特征值）可以定义该空间。
- 需要注意的是，不管k 取值如何，矩阵C_k 仍然是一个$M \times N$ 的矩阵。

示例1：词项-文档矩阵的SVD分解 低秩表示空间

例 18-4 考虑如下词项–文档矩阵 $C=$

	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

矩阵 U 为

	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

$\Sigma_2=$

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

奇异值矩阵Σ =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

计算 $C_2=$

	d_1	d_2	d_3	d_4	d_5	d_6
1	−1.62	−0.60	−0.44	−0.97	−0.70	−0.26
2	−0.46	−0.84	−0.30	1.00	0.35	0.65
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

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

示例1：词项-文档矩阵的SVD分解 低秩表示空间

截断 SVD 分解中的文档矩阵$(V')^T$为

	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

空间的维度从5降到了2

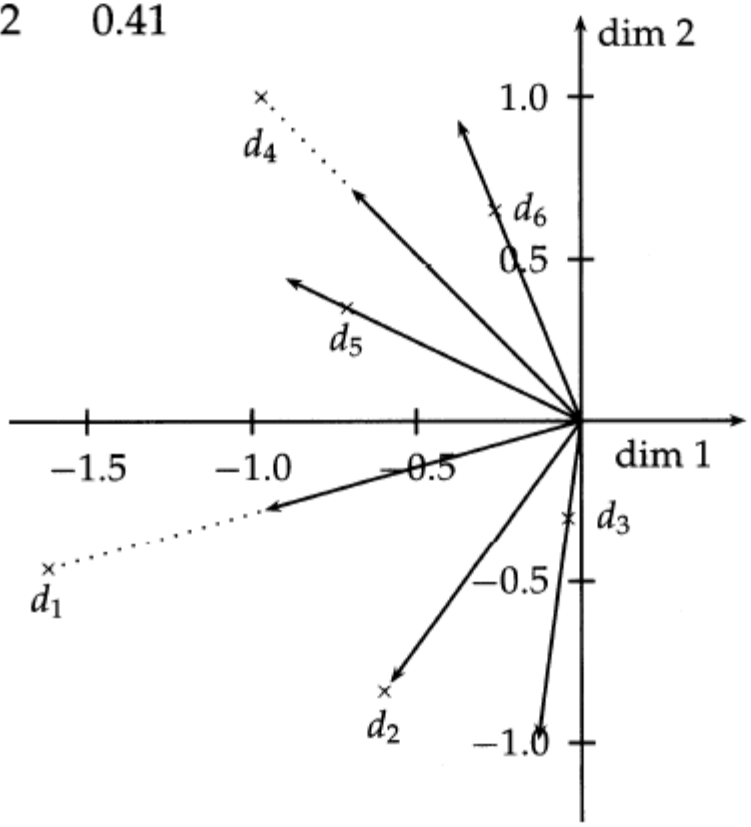


图 18-3　18-4 示例中文档简化成$(V')^T$中 2 维向量后的示意图

示例2：词项-文档矩阵的SVD分解
发现相关文档

c1:	*Human* machine *interface* for Lab ABC *computer* applications
c2:	A *survey of user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user*-perceived *response time* to error measurement
m1:	The generation of random, binary, unordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors: A survey*

文档集

原始的Term-Document矩阵

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Hars
Indexing By Latent Semantic Analysis. Journal of the American Society For Information Science, 41, 391-407. 16

示例2：词项-文档矩阵的SVD分解
发现相关文档

$T_0 =$

0.22	−0.11	0.29	−0.41	−0.11	−0.34	0.52	−0.06	−0.41
0.20	−0.07	0.14	−0.55	0.28	0.50	−0.07	−0.01	−0.11
0.24	0.04	−0.16	−0.59	−0.11	−0.25	−0.30	0.06	0.49
0.40	0.06	−0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	−0.17	0.36	0.33	−0.16	−0.21	−0.17	0.03	0.27
0.27	0.11	−0.43	0.07	0.08	−0.17	0.28	−0.02	−0.05
0.27	0.11	−0.43	0.07	0.08	−0.17	0.28	−0.02	−0.05
0.30	−0.14	0.33	0.19	0.11	0.27	0.03	−0.02	−0.17
0.21	0.27	−0.18	−0.03	−0.54	0.08	−0.47	−0.04	−0.58
0.01	0.49	0.23	0.03	0.59	−0.39	−0.29	0.25	−0.23
0.04	0.62	0.22	0.00	−0.07	0.11	0.16	−0.68	0.23
0.03	0.45	0.14	−0.01	−0.30	0.28	0.34	0.68	0.18

$S_0 =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

保留S_0的最大两个奇异值

$$C = U\Sigma V^T$$

$$X = T_0 S_0 D_0$$

$D_0 =$

0.20	−0.06	0.11	−0.95	0.05	−0.08	0.18	−0.01	−0.06
0.61	0.17	−0.50	−0.03	−0.21	−0.26	−0.43	0.05	0.24
0.46	−0.03	0.21	0.04	0.38	0.72	−0.24	0.01	0.02
0.54	−0.23	0.57	0.27	−0.21	−0.37	0.26	−0.02	−0.08
0.28	0.11	−0.51	0.15	0.33	0.03	0.67	−0.06	−0.26
0.00	0.19	0.10	0.02	0.39	−0.30	−0.34	0.45	−0.62
0.01	0.44	0.19	0.02	0.35	−0.21	−0.15	−0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	−0.03	−0.60	0.36	−0.04	−0.07	−0.45

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R.(1990). Indexing By Latent Semantic Analysis. Journal of the American Society For Information Science, 41, 391-407. 10

示例2：词项-文档矩阵的SVD分解

发现相关文档：根据SVD结果重构的$\hat{X}$

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	0
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	0
minors	0	0	0	0	0	0	0	1	0

原始的Term-Document矩阵X

X中human-C2值为0，因为C2中并不包含human单词，但是$\hat{X}$中human-C2为0.40，表明human和C2有一定的关系，为什么呢？因为C2: A survey of user opinion of computer system response time中包含user单词，和human是近似词，故human-C2的值被提高了。

保留S_0的最大两个奇异值重构的$\hat{X}$

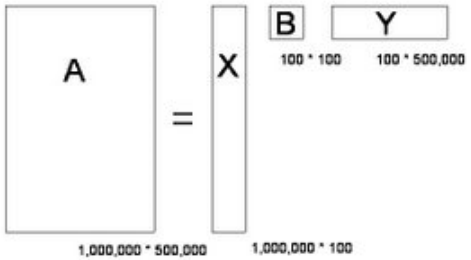
$\hat{X}$ =

0.16	0.40	0.38	0.47	0.18	−0.05	−0.12	−0.16	−0.09
0.14	0.37	0.33	0.40	0.16	−0.03	−0.07	−0.10	−0.04
0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
0.45	1.23	1.05	1.27	0.56	−0.07	−0.15	−0.21	−0.05
0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
0.22	0.55	0.51	0.63	0.24	−0.07	−0.14	−0.20	−0.11
0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
−0.06	0.23	−0.14	−0.27	0.14	0.24	0.55	0.77	0.66
−0.06	0.34	−0.15	−0.30	0.20	0.31	0.69	0.98	0.85
−0.04	0.25	−0.10	−0.21	0.15	0.22	0.50	0.71	0.62

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R.(1990). Indexing By Latent Semantic Analysis. Journal of the American Society For Information Science, 41, 391-407. 10

示例3：文档-词项矩阵SVD分解
词项、文档的聚类

奇异值分解就是把上面这样一个大矩阵，分解成三个小矩阵相乘，如下图所示。比如把上面的例子中的矩阵分解成一个一百万乘以一百的矩阵X，一个一百乘以一百的矩阵B，和一个一百乘以五十万的矩阵Y。这三个矩阵的元素总数加起来也不过1.5亿，仅仅是原来的三千分之一。相应的存储量和计算量都会小三个数量级以上。



三个矩阵有非常清楚的物理含义。第一个矩阵X中的每一行表示**意思相关的一类词**，其中的每个非零元素表示这类词中每个词的重要性（或者说相关性），数值越大越相关。最后一个矩阵Y中的每一列表示**同一主题一类文章**，其中每个元素表示这类文章中每篇文章的相关性。中间的矩阵则表示**类词和文章之间的相关性**。因此，我们只要对关联矩阵A进行一次奇异值分解，我们就可以同时完成了近义词分类和文章的分类。

示例3：词项-文档矩阵SVD分解

词项、文档的聚类

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

- 左奇异向量表示词的一些特性，右奇异向量表示文档的一些特性，中间的奇异值矩阵表示左奇异向量的一行与右奇异向量的一列的重要程序，数字越大越重要。
- 左奇异向量的第一列表示每一个词的出现频繁程度（是一个大概的描述），如book是0.15对应文档中出现的2次，investing是0.74对应了文档中出现了9次，rich是0.36对应文档中出现了3次；
- 右奇异向量中一的第一行表示每一篇文档中的出现词的个数的近似，比如说，T6是0.49，出现了5个词，T2是0.22，出现了2个词。

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

*

3.91	0	0
0	2.61	0
0	0	2.00

*

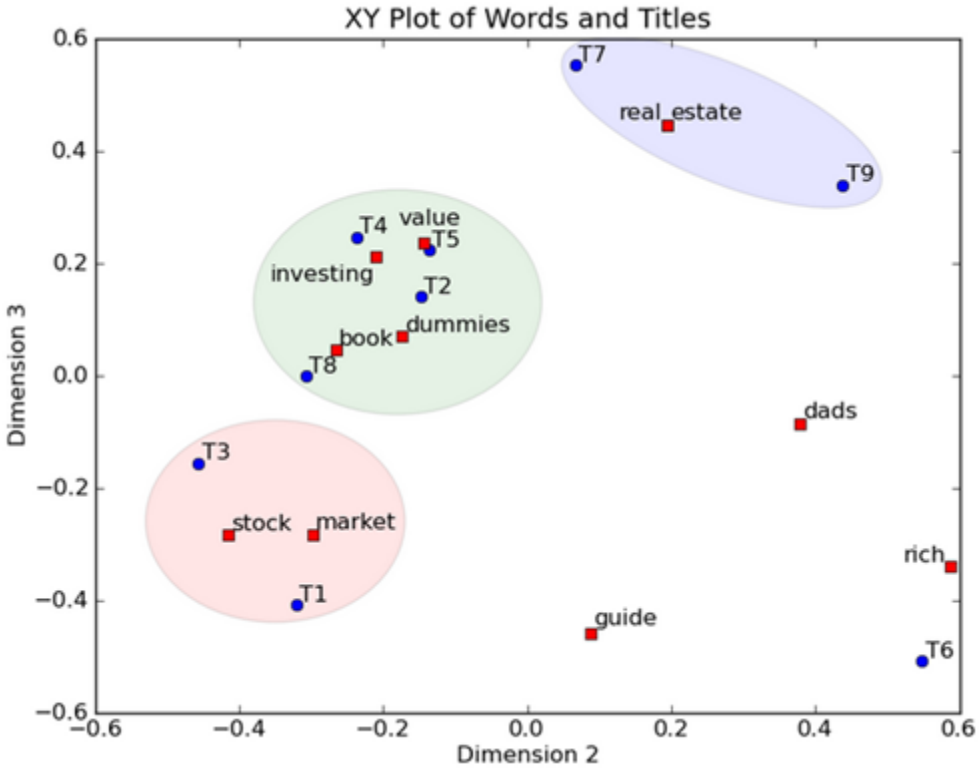
T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

http://www.cnblogs.com/liangflying/archive/2012/09/25/2701148.html

示例3：文档-词项矩阵SVD分解

词项、文档的聚类

将左奇异向量和右奇异向量都取后2维（之前是3维的矩阵），投影到一个平面上，可以得到右图。



book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

*

3.91	0	0
0	2.61	0
0	0	2.00

*

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

小结：LSI / LDA

Scott Deerwester (born 1956) is one of the inventors of latent semantic analysis. He was a member of the faculty of the Colgate University, University of Chicago and the Hong Kong University of Science and Technology. He has been resident in Hong Kong since 1991, where he has been working in the humanitarian sector in recent years.

- 词项-文档矩阵
  - 一义多词（*synonymy*）问题
  - 一词多义（*polysemy*）问题
- 隐性语义索引
  - LSI (Latent Semantic Indexing)
  - LDA (Latent Semantic Analysis)
  - 文档和查询映射到这个低秩表示空间

- 将词项-文档矩阵的低秩逼近与IR进行关联的工作来自Deerwester 等人（1990），后来的相关结果的综述参见Berry 等人（1995）。Dumais（1993）及 Dumais（1995）介绍了他们在TREC上的实验，其结果表明，至少在某些基准测试上LSI所产生的结果的正确率和召回率会高于常规的向量空间检索方法。

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

文档-词项矩阵

用一个大矩阵A来描述这一百万篇文章和五十万词的关联性。这个矩阵中，每一行对应一篇文章，每一列对应一个词。

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1j} & a_{1N} \\ \cdots & & & \cdots \\ a_{i1} & & a_{ij} & a_{iN} \\ \cdots & & & \cdots \\ a_{M1} & \cdots & a_{Mj} & a_{MN} \end{pmatrix}$$

M=1,000,000，N=500,000。第 i 行，第 j 列的元素，是字典中第 j 个词在第 i 篇文章中出现的加权词频（比如，TF/IDF)。这个矩阵非常大，有一百万乘以五十万，即五千亿个元素。

如何用计算机进行奇异值分解

- 现在剩下的唯一问题，就是如何用计算机进行奇异值分解。这时，线性代数中的许多概念，比如矩阵的特征值等等，以及数值分析的各种算法就统统用上了。
- 并行计算
  - 在很长时间内，奇异值分解都无法并行处理。（虽然Google 早就有了MapReduce 等并行计算的工具，但是由于奇异值分解很难拆成不相关子运算，即使在 Google 内部以前也无法利用并行计算的优势来分解矩阵。）最近，Google 中国的张智威博士和几个中国的工程师及实习生已经实现了奇异值分解的并行算法，我认为这是Google 中国对世界的一个贡献。

吴军《数学之美》

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

何谓推荐系统？

- 大多数大规模的商业和社交网站都会向用户推荐选项，比如产品或要联系的人。推荐引擎对大量数据进行分类，以识别潜在的用户偏好。
- 推荐系统改变了没有活力的网站与其用户通信的方式。无需提供一种静态体验，让用户搜索并可能购买产品，推荐系统加强了交互，以提供内容更丰富的体验。推荐系统根据用户过去的购买和搜索历史，以及其他用户的行为，自主地为各个用户识别推荐内容。
- 基本方法：大多数推荐系统都采用两种基本方法之一：协作式过滤或基于内容的过滤。当然也有其他的方法（比如混合方法）。

http://www.ibm.com/developerworks/cn/views/web/libraryview.jsp

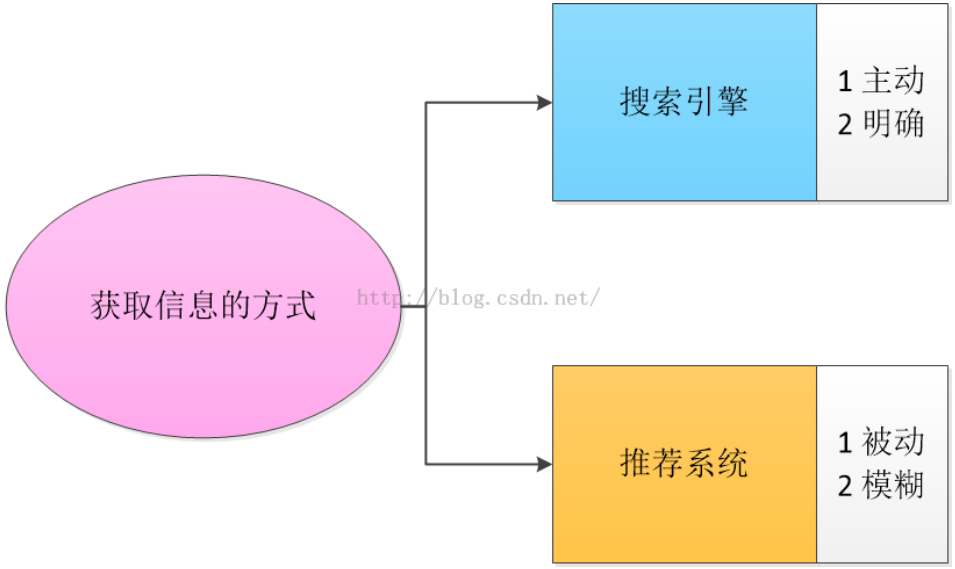
典型的推荐系统

- LinkedIn（面向业务的社交网络站点）推荐您可能知道的人，您可能喜欢的工作，您想要关注的群组，或者您可能感兴趣的公司。LinkedIn 使用 Apache Hadoop 构建它的协作式过滤功能。
- Amazon（流行的电子商务站点）使用基于内容的推荐。您选择一款要购买的商品后，Amazon 根据该原始商品来推荐其他用户已购买的商品（作为下一个可能购买的商品表格）。Amazon 为此行为申请了专利，称为商品到商品协作式过滤 (item-to-item collaborative filtering)。
- Hulu（一个流视频网站）使用一个推荐引擎识别用户可能感兴趣的内容。它还在线下使用基于商品的协作式过滤和 Hadoop 来扩展对大量数据的处理。
- Netflix（一个视频租赁和流式传输服务）是一个知名的例子。
  - 2006 年，Netflix 举办了一次竞赛以改进它的推荐系统 Cinematch。
  - Netflix竞赛有效地推动了学术界和产业界对推荐算法的研究。
- 其他整合了推荐引擎的站点包括 Facebook、Twitter、Google、MySpace、Last.fm、Del.icio.us、Pandora、Goodreads，……

http://www.ibm.com/developerworks/cn/views/web/libraryview.jsp

搜索引擎（Search Engine）
推荐系统（Recommendation System）

从信息获取的角度来看，搜索和推荐是用户获取信息的两种主要手段。搜索是一个非常主动的行为，并且用户的需求十分明确，在搜索引擎提供的结果里，用户也能通过浏览和点击来明确的判断是否满足了用户需求。然而，推荐系统接受信息是被动的，需求也都是模糊而不明确的。



国际知识发现和数据挖掘竞赛（KDD-CUP）
KDD: Knowledge Discovery and Data Mining

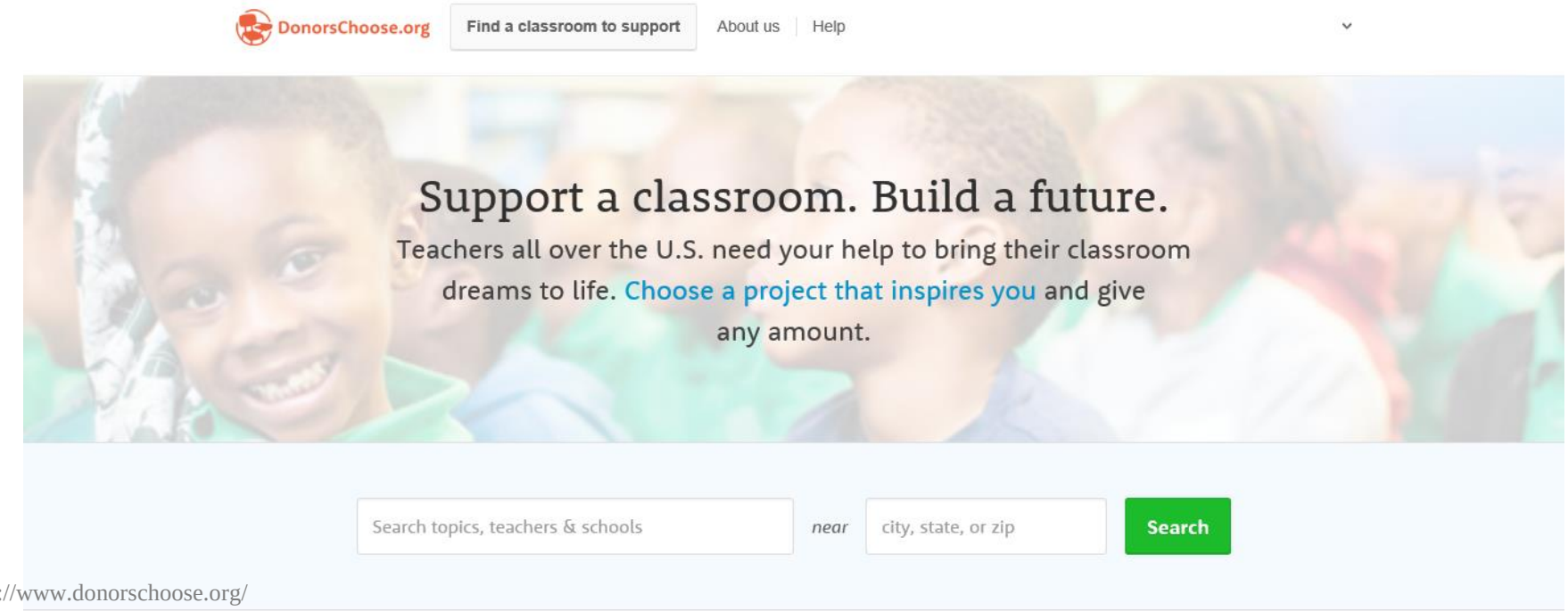
- 竞赛是由ACM 的数据挖掘及知识发现专委会（SIGKDD）主办
  - KDD Cup 2019： Auto-ML
  - KDD Cup 2018: 环境数据预测
  - KDD Cup 2017(阿里): Highway Tollgates Traffic Flow Prediction
  - KDD Cup 2016 (微软): Whose papers are accepted the most
  - KDD Cup 2015 (学堂在线) ： Predicting dropouts in MOOC
  - KDD Cup 2014 (DonorsChoose) ： Predicting Excitement at DonorsChoose
  - KDD Cup 2013 (微软): Determine whether an author has written a given paper
  - KDD Cup 2012(腾讯)
    - Track1任务：社交网络中的个性化推荐系统
    - Track2任务：搜索广告系统的pTCR点击率预估
  - KDD Cup 2011(雅虎)
    - Track1任务：音乐评分预测
    - Track2任务：识别音乐是否被用户评分
  - KDD Cup 2010 (卡耐基梅隆大学)
    - 根据智能教学辅导系统和学生之间的交互日志，来预测学生数学题的考试成绩。
  - KDD Cup 2009 (法国电信Orange)
    - 根据法国电信运营商的数据预测客户三个维度的属性：忠诚度、购买欲、增值性
  - KDD Cup 2008(西门子医疗) ： Breast cancer
  - KDD-Cup 2007(Netflix) ： Consumer recommendations

这个阶段推荐系统引起广泛关注

2014 KDD Cup (DonorsChoose)

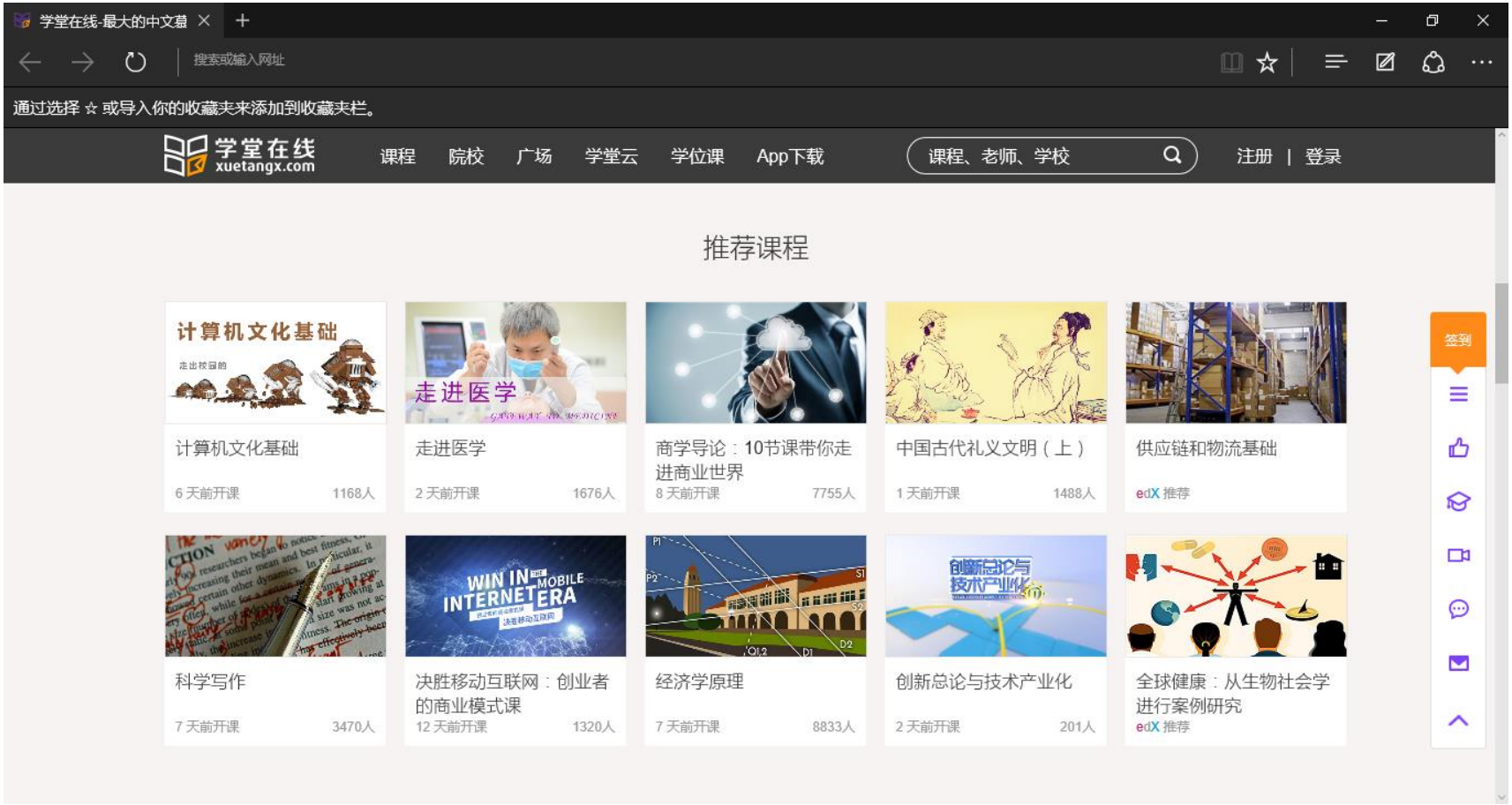
Predicting Excitement at DonorsChoose.org

The 2014 KDD Cup asks participants to help DonorsChoose.org identify projects that are exceptionally exciting to the business, at the time of posting. While all projects on the site fulfill some kind of need, certain projects have a quality above and beyond what is typical. By identifying and recommending such projects early, they will improve funding outcomes, better the user experience, and help more students receive the materials they need to learn.



http://www.donorschoose.org/
https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose/

2015 KDD Cup (学堂在线)
Predicting dropouts in MOOC



学堂在线是由清华大学研发出的中文MOOC（大规模开放在线课程，简称慕课）平台，于2013年10月10日正式启动

http://www.kddcup2015.com

KDD Cup 2016 (微软)

Whose papers are accepted the most: towards measuring the impact of research institutions

https://kddcup2016.azurewebsites.net/

2016年KDD Cup竞赛题目是“谁的论文被录用最多：研究机构影响力度量”。组织者指定8个计算机科学不同领域的顶级会议，参赛队伍通过分析微软学术网络（Microsoft Academic Graph）数据集和其他开源数据，预测今年在这些会议上各个研究机构发表论文情况的排名。与往届不同，今年KDD Cup的特点是在比赛结束之前组织者和参赛者都不知道会议的真实论文录取情况，参赛者需要自己设计并评价预测算法，这使比赛更加有趣和富有挑战。

KDD Cup 2017

https://tianchi.aliyun.com/

Highway Tollgates Traffic Flow Prediction: Travel Time & Traffic Volume Prediction

源自阿里云人工智能ET在交通领域实施的案例之一，需要选手基于历史数据预测高速路口收费站的流量和通行时间。

KDD Cup2017优胜者

https://www.leiphone.com/news/201708/wej6EvvxqR58YrK5.html

Travel Time Prediction | Volume Prediction | Travel Time Prediction | Volume Prediction

排名	参赛者	所在组织
1	Convolution	Microsoft
2	好想有个队友	浙江大学
3	一个师的兵力	釧路公立大学
4	Pseudo_Code_vol2	国立台湾科技大学
5	萌萌哒の小云	东南大学
6	inplus	中山大学
7	INNOVA-TSN	Innova-tsn
8	jps jps	名寄市立大学
9	潘神的小跟班	上海财经
10	好名字	武汉大学
11	EUSKOTALDEA	UCM
12	好啊好啊	北京科技大学
13	InfiniteWing	国立中正大学

排名	参赛者	所在组织
1	Convolution	Microsoft
2	Black-Swan	JDDog
3	成交量遥遥领先	瓜子二手车(guazi.com
4	一个师的兵力	釧路公立大学
5	我想想。。。	星火智慧
6	xia95	lbr
7	Multi-task专家	北京航空航天大学
8	Why	jd.com
13	FNo000000	北京科技大学

2017年8月13-17日，第23届KDD大会在加拿大哈利法克斯召开。微软、美团联合北京航空航天大学的Convolution 团队在KDD Cup 2017的两项任务中包揽第一

KDD Cup 2018

https://biendata.com/competition/kdd_2018/

给出北京、伦敦各空气监测站点2017-2018年每天每小时的空气质量监测数据，以及当时附近的环境质量数据，预测未来48小时内北京35个站点的PM2.5，PM10和O3的浓度，以及伦敦13个站点的PM2.5和PM10的浓度，预测数据将与未来的实时数据进行比较（连续21天预测数据的经验风险）。
KDD CUP 2018设立三项大奖，分别为 General Track、Last Ten-Day Prediction Track 以及 Second 24-Hour Prediction Track，从不同维度奖励表现突出的团队。

主要奖项,一等奖（10,000美金）：中南大学 Haoran Jiang, Binli Luo；北京邮电大学 Jindong Han, Juan Liu, Qianqian Zhang PPT
最后10天专项奖,一等奖（5,000美金）：微软Zhipeng Luo；北京大学 Jianqiang Huang；阿里巴巴Ke Hu
最佳长期预测奖,一等奖（2,500美金）：微软Zhipeng Luo；北京大学 Jianqiang Huang；阿里巴巴Ke Hu

KDD Cup 2019

KDD Cup 2019三项重大赛事：Auto-ML Track、Regular ML Track及Humanity RL Track。AutoML（Automated/Automatic Machine Learning，自动机器学习）旨在研究在没有专业知识的情况下、使用的低门槛甚至零门槛的机器学习算法，在AI人才紧缺的情况下，AutoML可以降低AI落地过程中对科学家的依赖。

据悉，本次KDD Cup AutoML挑战赛由国内AI公司第四范式主办，微软、AutoML领域学术组织ChaLearn协办，并为此次比赛设置了比赛项目——基于时序关系型数据的AutoML。

国际知识发现和数据挖掘竞赛（KDD-CUP）
KDD: Knowledge Discovery and Data Mining

- KDD Cup 2006, data mining for medical diagnosis, specifically identifying pulmonary embolisms from three-dimensional computed tomography data.
- KDD Cup 2005, Internet user search query classification.
- KDD Cup 2004, features tasks in particle physics and bioinformatics evaluated on a variety of different measures.
- KDD Cup 2003, focuses on problems motivated by network mining and the analysis of usage logs.
- KDD Cup 2002, focus: bioinformatics and text mining
- KDD Cup 2001, focus: bioinformatics and drug discovery.
- KDD Cup 2000, focus: web mining tasks.
- KDD Cup 1999, focus: intrusion detection and report
- KDD Cup 1998, focus: direct marketing, list with best donation value; best report
- KDD Cup 1997, focus: predicting most likely donors for a charity

搜索引擎中融合的推荐系统元素

小结：推荐系统的兴起

- 1996年， Yahoo网站推出了个性化入口MyYahoo，可以看作第一个正式商用的推荐系统。
- Amazon网站
- 2006年，Netflix Prize
- 2011百度世界大会，李彦宏将推荐引擎与云计算、搜索引擎并列为未来互联网重要战略规划以及发展方向。

- ACM SIGIR自2001年起把推荐系统作为该会议的独立主题
- ACM RecSys (ACM Recommender Systems Conference)始于2007

- 国际知识发现和数据挖掘竞赛（KDD-CUP）
  - KDD: Knowledge Discovery and Data Mining
  - KDD CUP 2011 : “音乐评分预测”和“识别音乐是否被用户评分”
  - KDD CUP 2012 : “微博中的好友推荐”和“计算广告中的点击率预测”

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

基本方法1：基于内容的过滤（Content-based Recommendation）

- 基于内容的过滤：可根据用户的行为来构造推荐内容。例如，此方法可能使用历史浏览信息，比如用户阅读了哪些博客和这些博客的特征。如果用户经常阅读关于 Linux 的文章或可能在有关软件工程的博客中留下了评论，基于内容的过滤可使用此历史信息来识别和推荐类似的内容（有关 Linux 的文章或有关软件工程的其他博客）。可手动定义此内容，或者根据其他类似性方法来自动提取此内容。

http://www.ibm.com/developerworks/cn/opensource/os-recommender1/index.html
http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html

基本方法2：协作式过滤（Collaborative Filtering Recommendation）

- 协作式过滤根据以前用户行为的模型来获得推荐内容。可通过单个用户的行为单独构造该模型，或者，更有效的方法是，根据其他拥有类似特征的用户的行为来构造该模型。考虑其他用户的行为时，协作式过滤使用群组知识并基于类似用户来形成推荐内容。在本质上，推荐内容**基于多个用户的自动协作，并过滤出表现了类似偏好或行为的用户**。
- 例如，假设您正在构建一个网站来推荐博客。通过使用许多订阅并阅读博客的用户的信息，您可根据这些用户的偏好将他们分组。例如，您可将阅读多篇相同博客的用户分组到一起。有了此信息，您可识别该群组阅读了哪些最流行的博客。然后 — 对于群组中的一个特定用户 — 您推荐他或她未阅读也未订阅的最流行博客。

http://www.ibm.com/developerworks/cn/opensource/os-recommender1/index.html
http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html

协作式过滤的简单示例

图中博客行和用户列相交的单元包含该博客的该用户所阅读的文章数量。通过根据用户的阅读习惯来为用户划分集群，这里有两个用户的集群。每个集群的成员的阅读习惯相似：Marc 和 Elise（都阅读了多篇关于 Linux 和云计算的文章）形成 Cluster 1。Cluster 2 中包含 Megan 和 Jill，他们都阅读了多篇关于 Java和敏捷性的文章。

图 1. 协作式过滤的简单示例

Blogs	Marc	Megan	Elise	Jill
Linux	13	3	11	-
OpenSource	10	-	-	3
Cloud Computing	6	1	9	-
Java Technology	-	6	-	9
Agile	-	7	1	8
		Articles read per user		
Cluster	1	2	1	2

在 Cluster 1 中，Marc 阅读了 10 篇开源博客文章，Elise 一篇都没读；Elise 阅读了 1 篇敏捷性博客，Marc 一篇都没读。故针对 Elise 的一个推荐内容是开源博客；无法对 Marc 进行推荐。

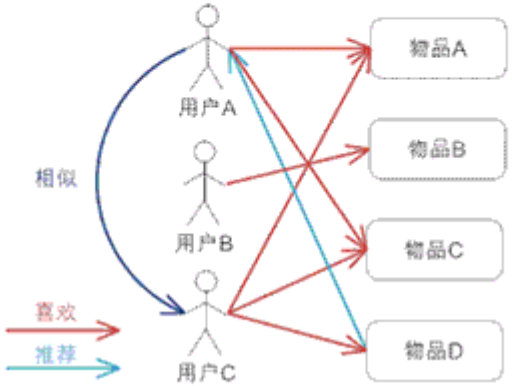
在 Cluster 2 中，Jill 阅读了 3 篇开源博客，而 Megan 一篇都没读；Megan 阅读了 11 篇 Linux 博客，而 Jill 一篇都没读。故为 Jill 推荐 Linux 博客和为 Megan 推荐开源博客。

http://www.ibm.com/developerworks/cn/opensource/os-recommender1/index.html
http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html

基于用户的 CF（User CF）

基本思想：基于用户对物品的偏好找到相邻邻居用户，然后将邻居用户喜欢的推荐给当前用户。计算上，就是将一个用户对所有物品的偏好作为一个向量来计算用户之间的相似度，找到 K 邻居后，根据邻居的相似度权重以及他们对物品的偏好，预测当前用户没有偏好的未涉及物品，计算得到一个排序的物品列表作为推荐。图中，对于用户 A，根据用户的历史偏好，这里只计算得到一个邻居 - 用户 C，然后将用户 C 喜欢的物品 D 推荐给用户 A。

用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√

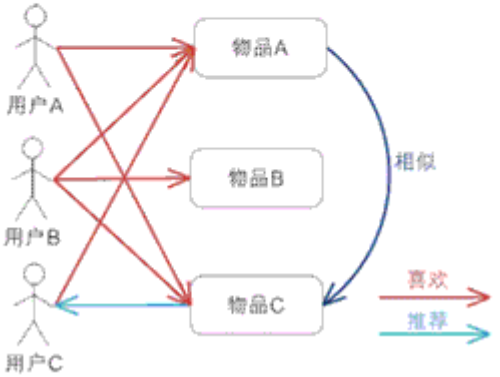


http://www.ibm.com/developerworks/cn/opensource/os-recommender1/index.html
http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html

基于物品的 CF（Item CF）

基本思想：在计算邻居时采用物品本身，而不是从用户的角度，即基于用户对物品的偏好找到相似的物品，然后根据用户的历史偏好，推荐相似的物品给他。从计算的角度看，就是将所有用户对某个物品的偏好作为一个向量来计算物品之间的相似度，得到物品的相似物品后，根据用户历史的偏好预测当前用户还没有表示偏好的物品，计算得到一个排序的物品列表作为推荐。图中，对于物品 A，根据所有用户的历史偏好，喜欢物品 A 的用户都喜欢物品 C，得出物品 A 和物品 C 比较相似，而用户 C 喜欢物品 A，那么可以推断出用户 C 可能也喜欢物品 C。

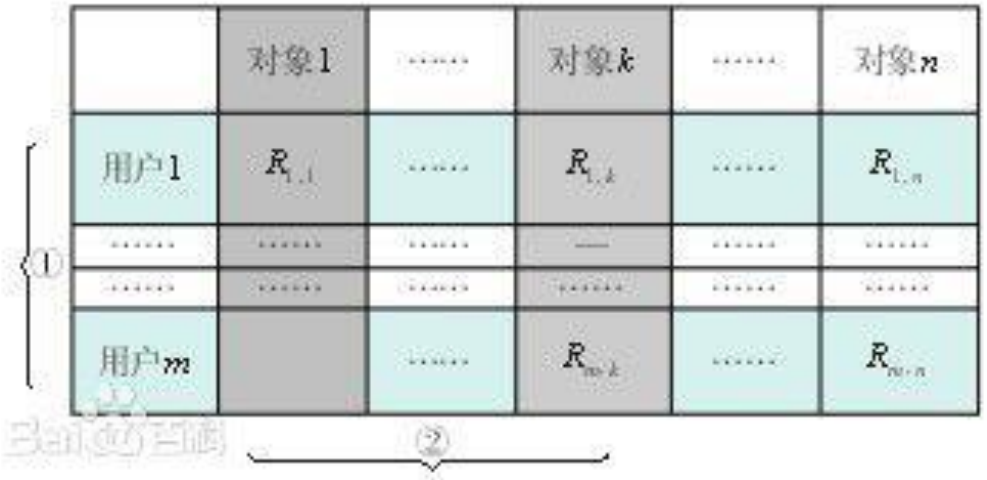
用户/物品	物品A	物品B	物品C
用户A	√		√
用户B	√	√	√
用户C	√		推荐



http://www.ibm.com/developerworks/cn/opensource/os-recommender1/index.html
http://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy2/index.html

小结：推荐系统的基本方法

- 效用矩阵（或评分矩阵）的填充
  - Utility Matrix: user, item



- 基于内容（Content-based）
  - 关注项的属性：考虑物品之间的相似性
- 协同过滤（Collaborative Filtering）
  - 关注用户和项之间的关系，不考虑物品本身的属性

矩阵分解在信息检索中的应用

- 矩阵分解及隐性语义索引
  - 关于词项-文档矩阵
  - 线性代数基础
  - 矩阵分解与低秩逼近
  - IR中的隐性语义索引
  - 矩阵分解的计算机实现
- 推荐系统
  - 推荐系统的兴起
  - 推荐系统的基本方法
  - 示例：UV分解用于音乐推荐

示例：UV分解用于音乐推荐

量化标准：单曲循环=5, 分享=4, 收藏=3, 主动播放=2 , 听完=1, 跳过=-2 , 拉黑=-5，在分析时能获得的实际评分矩阵R，也就是输入矩阵大概是这个样子：

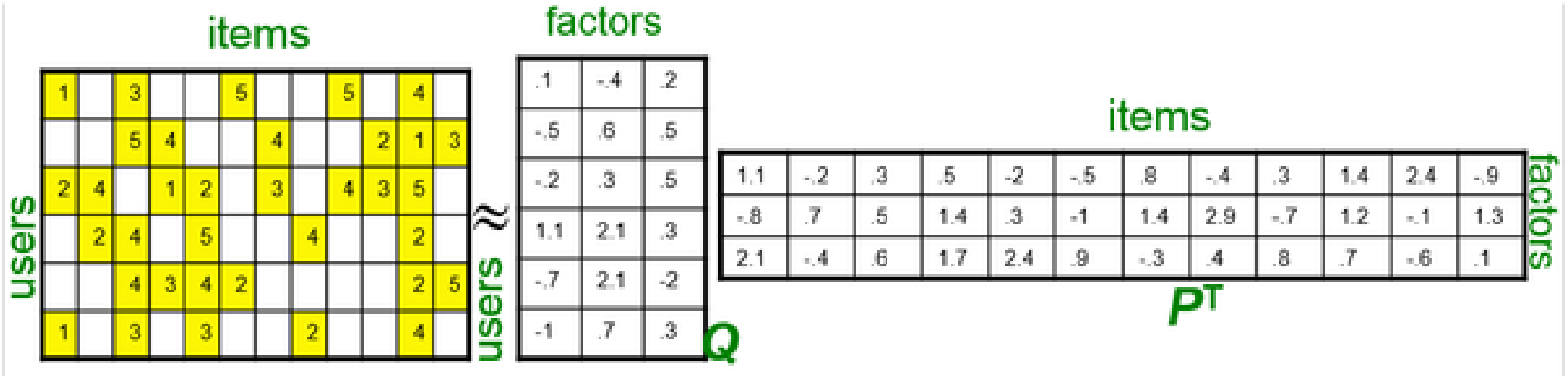
	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

这是个非常非常稀疏的矩阵

推荐系统的目标就是预测出“？”对应位置的分值。

示例：UV分解用于音乐推荐

矩阵的UV分解：将上面的评分矩阵分解为两个低维度的矩阵，用Q和P两个矩阵的乘积去估计实际的评分矩阵，而且我们希望估计的评分矩阵$\tilde{R}$



优化问题的目标函数：$min_{P,Q}\Sigma(r_{ui} - q_i p_u^T)^2$

示例：UV分解用于音乐推荐

UV分解的结果

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

	因子1	因子2	因子3	因子4	因子5
用户1	0.908	0.642	0.524	0.454	0.406
用户2	0.877	0.620	0.506	0.438	0.392
用户3	0.768	0.543	0.443	0.384	0.344
用户4	0.853	0.603	0.492	0.426	0.381
用户5	0.847	0.599	0.489	0.424	0.379
用户6	0.884	0.625	0.510	0.442	0.395
用户7	0.870	0.615	0.502	0.435	0.389
用户8	0.878	0.621	0.507	0.439	0.392

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	0.914	0.913	0.906	0.921	0.850	0.900	0.919	0.937	0.931	0.947	0.891	0.937	0.900
用户2	0.646	0.645	0.640	0.652	0.601	0.636	0.650	0.663	0.658	0.670	0.630	0.663	0.636
用户3	0.528	0.527	0.523	0.532	0.491	0.520	0.531	0.541	0.537	0.547	0.514	0.541	0.520
用户4	0.457	0.456	0.453	0.461	0.425	0.450	0.460	0.469	0.465	0.473	0.445	0.469	0.450
用户5	0.409	0.408	0.405	0.412	0.380	0.402	0.411	0.419	0.416	0.423	0.398	0.419	0.402
用户6													
用户7													
用户8													

示例：UV分解用于音乐推荐

两个矩阵相乘就可以得到估计的得分矩阵

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1	5					-5			5	3		1	5
用户2				3					3				4
用户3			1		2	-5	4			-2	-2		-2
用户4		4	4	3			-2		-5			3	
用户5		5	-5		-5		4	3			4		
用户6			4			3			4				
用户7		-2				5				4		4	-2
用户8		-2				5		5		4			-2

	音乐1	音乐2	音乐3	音乐4	音乐5	音乐6	音乐7	音乐8	音乐9	音乐10	音乐11	音乐12	音乐13
用户1		2.10	2.08	2.12	1.96		2.12	2.16			2.05		
用户2	2.03	2.03	2.01		1.89	2.00	2.04	2.08		2.10	1.98	2.08	
用户3	1.78	1.78		1.80				1.83	1.82			1.83	
用户4	1.98				1.84	1.95		2.03		2.05	1.93		1.95
用户5	1.96			1.98		1.93			2.00	2.04		2.01	1.93
用户6	2.05	2.04		2.06	1.90		2.06	2.10		2.12	2.00	2.10	2.02
用户7	2.02		2.00	2.03	1.87		2.03	2.07	2.05		1.96		
用户8	2.03		2.01	2.05	1.89		2.04		2.07		1.98	2.09	

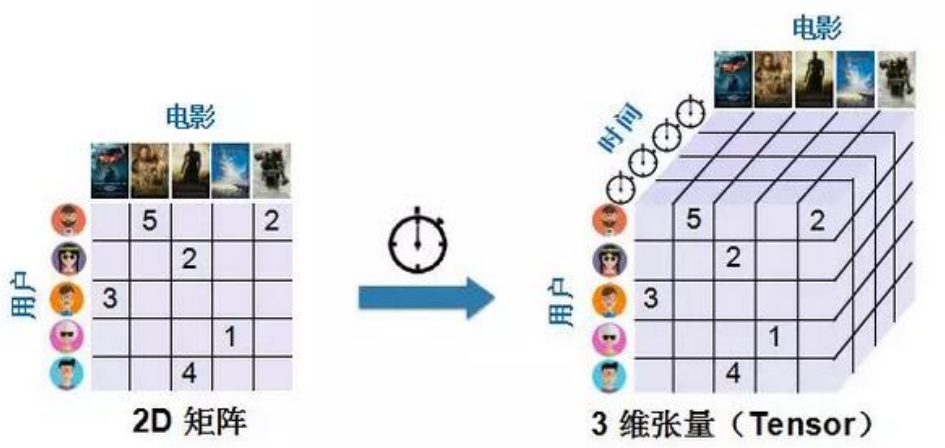
小结：效用矩阵填充的实现

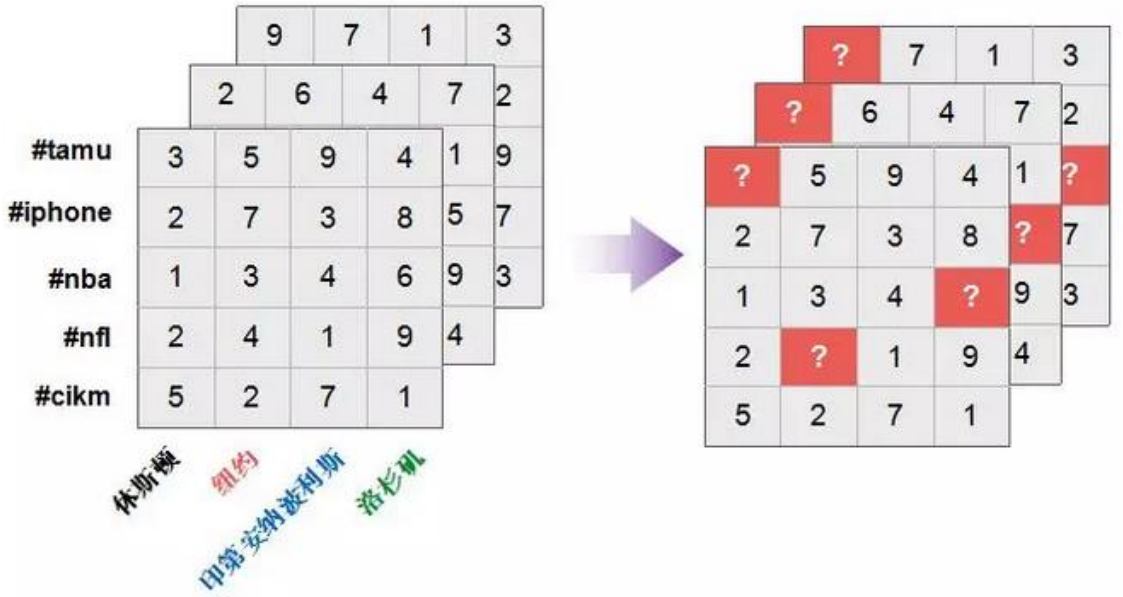
- 一种实现思路：M=UV，首先假设U、V矩阵内全为1,得到一个M_1,计算M和M1的均方根误差(RMSE,root meansquare error)；随机调整U或V中的值，重新构造M_2，再次计算其与原矩阵的RMSE(M,M_i)反复以上步骤,控制RMSE(M,M_i)越来越小，从而用M_i估计出M。

$$\boldsymbol{M}\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n_2-1} & ? \\ m_{21} & ? & & m_{2n_2-1} & m_{2n_2} \\ \vdots & & \ddots & & \vdots \\ m_{n_1-11} & & & & ? \\ ? & m_{n_1 2} & \cdots & m_{n_1 n_2-1} & m_{n_1 n_2} \end{bmatrix} = \boldsymbol{U}\begin{bmatrix} u_{11} & \cdots & u_{1d} \\ u_{21} & & u_{2d} \\ \vdots & \ddots & \vdots \\ u_{n_1 1} & \cdots & u_{n_1 d} \end{bmatrix} \times \boldsymbol{V}\begin{bmatrix} v_{11} & v_{12} & & v_{1n_2} \\ \vdots & & \ddots & \vdots \\ v_{d1} & v_{d2} & & v_{dn_2} \end{bmatrix}$$

补充：从矩阵到张量

张量(Tensor)的技术以及它在不同场景中的应用
https://blog.csdn.net/Mlooker/article/details/80492932





补充：Tensor Completion

Although the low rank approximation problem has been well studied for matrices, there is not much work on tensors, which are a higher dimensional extension of matrices. One major challenge lies in an appropriate definition of the trace norm for tensors. To the best of our knowledge, this has been not addressed in the literature. In this paper, we make two main contributions: 1) We lay the theoretical foundation of low rank tensor completion and propose the first definition of the trace norm for tensors. 2) We are the first to propose a solution for the low rank completion of tensors.

Liu, J., P. Musialski, P. Wonka and J. Ye (2013). "Tensor Completion for Estimating Missing Values in Visual Data." IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1): 208-220.

Tensor Completion

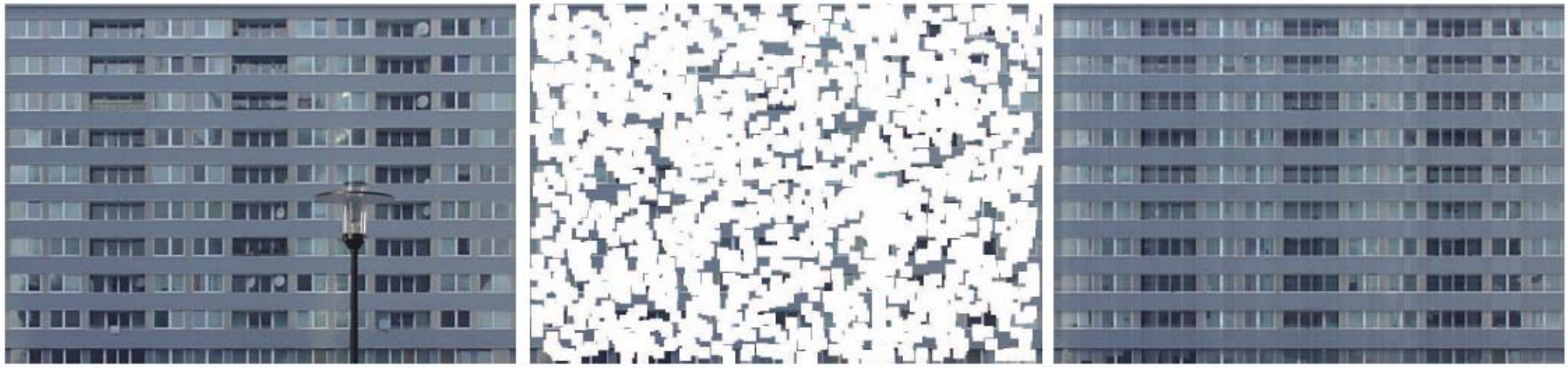


Fig. 7. Facade in-painting. The left image is the original image; we select the lamp and satellite dishes, together with a large set of randomly positioned squares, as the missing parts, shown in white in the middle image; the right image is the result of the proposed completion algorithm.

Fig. 8. Video completion. The left image (one frame of the video) is the original; we randomly select pixels for removal, shown in white in the middle image; the right image is the result of the proposed LTRC algorithm.

Liu, J., P. Musialski, P. Wonka and J. Ye (2013). "Tensor Completion for Estimating Missing Values in Visual Data." IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1): 208-220.

总结：矩阵分解在信息检索中的应用

- 信息检索系统
  - 词项-文档矩阵：布尔值、计数、权重
  - SVD分解与低秩逼近
  - IR中的隐性语义索引 ←同义词
- 推荐系统
  - User-Item矩阵（效用矩阵、评分矩阵）
  - 协同过滤（Collaborative Filtering） ←相似用户（项）
  - UV分解