

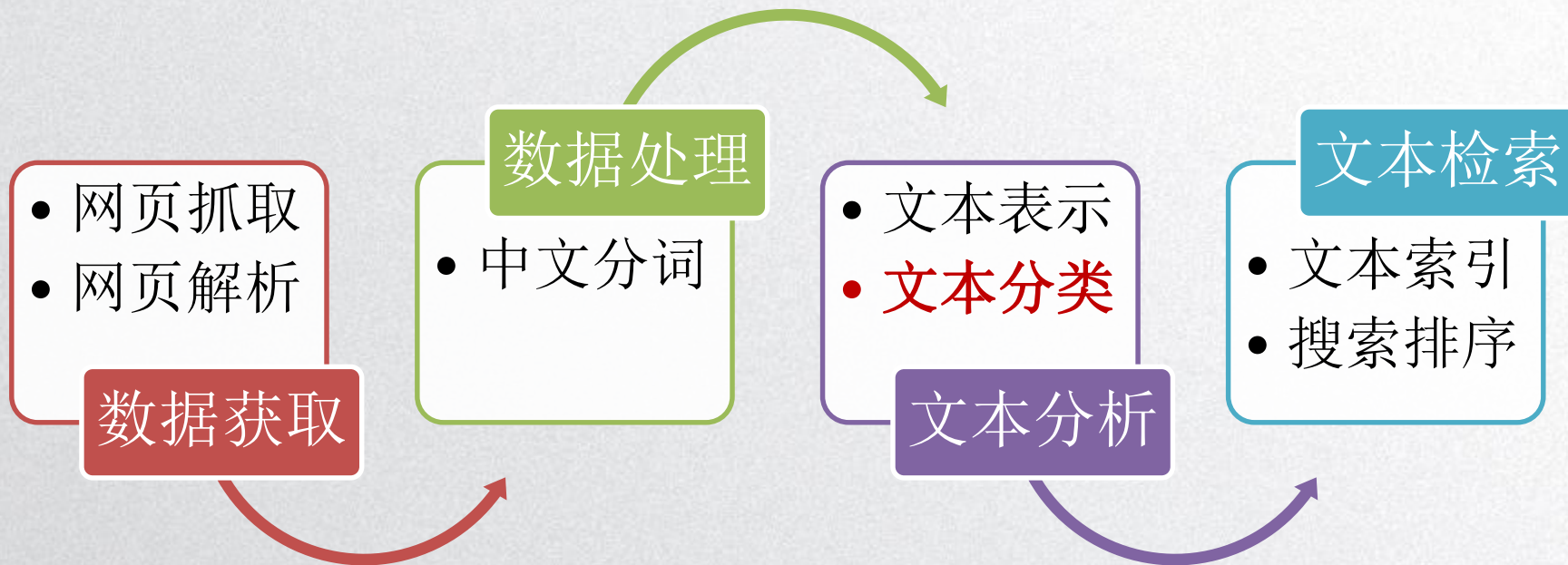


# 文本表示1：TF-IDF、SVD




覃雄派

# 文本模块涉及的内容



# 文本表示1: TF-IDF、SVD

- 什么是文本表示
  - 以关键词搜索场景为例
  - 考虑以下5篇文档, 与关键词查询 “dies, dagger”
    - 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
  - 你觉得文档d1和d4哪个与查询更相关?
  - 计算机能自动推理出这个吗?

A: d1

B: d4

# 提纲

- 什么是文本表示
- 独热编码(one hot encoding)
- 绝对词频
- TF-IDF
- 分布式表示入门
- LSI (LSA)



文本表示1: TF-IDF、  
SVD



# 文本表示1: TF-IDF、SVD

- 什么是文本表示

- 旨在寻求自然语言文本在语义层面更加**精练的和一致的表示形态**

- 单词层面**: 挖掘**单词间**隐含的语义关联关系
    - 查询/文档层**: 语义层面更加精准表示



分词后文本的形态:  
由单词组成的字符串

A cat sat on the mat

文本表示



	维度1	维度2	...	维度K
单词1				
单词2				
...				
单词N				

	维度1	维度2	...	维度K
文档1				
文档2				
...				
文档M				

- 文本表示的目的: 各种运算, 如: 相似度、距离等
  - 进而完成后续分析和处理



# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)
  - 假设有3个文档
  - 如何采用独热编码进行表示?

Doc1: I am a boy a boy.

Doc2: You are a girl a girl.

Doc3: We are different, different, different.



# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)

- 假设有3个文档
- 如何采用独热编码进行表示?

Doc1: I am a boy a boy.

Doc2: You are a girl a girl.

Doc3: We are different, different, different.

- (1) 建立字典表
- 字典表大小, 为不同的词项 (单词) 的个数

字典表	I	am	a	boy	you	are	girl	we	different
-----	---	----	---	-----	-----	-----	------	----	-----------

# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)

- 假设有3个文档
- 如何采用独热编码进行表示?

Doc1: I am a boy a boy.

Doc2: You are a girl a girl.

Doc3: We are different, different, different.

- (1) 建立字典表
- (2) 表示各个单词
  - 每个单词1个向量
  - 向量维数=字典表大小
  - 只有一个元素是1

字典表	I	am	a	boy	you	are	girl	we	different
I	1	0	0	0	0	0	0	0	0
am	0	1	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0
boy	0	0	0	1	0	0	0	0	0
you	0	0	0	0	1	0	0	0	0
are	0	0	0	0	0	1	0	0	0
girl	0	0	0	0	0	0	1	0	0
we	0	0	0	0	0	0	0	1	0
different	0	0	0	0	0	0	0	0	1



# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)

- 假设有3个文档
- 如何采用独热编码进行表示?

Doc1: I am a boy a boy.

Doc2: You are a girl a girl.

Doc3: We are different, different, different.

- (1) 建立字典表
- (2) 表示各个单词
- (3) 表示文档

字典表	I	am	a	boy	you	are	girl	we	different
doc1	1	1	1	1	0	0	0	0	0
doc2	0	0	1	0	1	1	1	0	0
doc3	0	0	0	0	0	1	0	1	1

# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding) 可以看作是一种集合表示
  - 集合中的元素: 出现在文本中的单词

文档



单词



	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							



集合

- 1, 如果规定单词的顺序, 这就是向量
- 2, 如果不规定单词的顺序, 这就是集合

# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)
  - 相似度计算: 基于集合的相似度计算, 如: Jaccard相似度
    - A和B的大小可以不一致
    - $0 \leq \text{JACCARD}(A, B) \leq 1$

文档



单词



	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							

$$\text{JACCARD}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$(A \neq \emptyset \text{ or } B \neq \emptyset)$

$\text{JACCARD}(A, A) = 1$

$\text{JACCARD}(A, B) = 0 \text{ if } A \cap B = 0$

jaccard\_百度翻译

jaccard

网络 杰卡德;

# 文本表示1: TF-IDF、SVD

- 独热编码 (One Hot Encoding)

- 基于集合的文本相似度计算

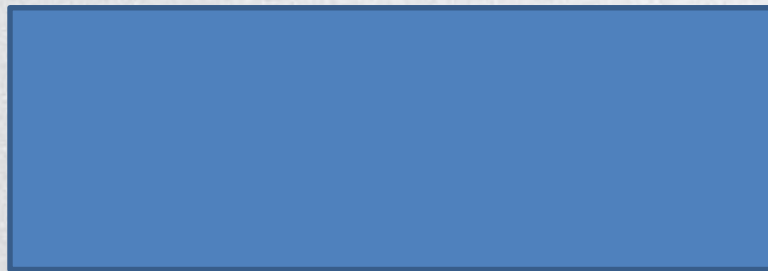
- D1: "ides of March"

- A = {ides, of, March}

- D2: "Caesar died in March"

- B = {Caesar, died, in, March}

$$\text{sim}(D1, D2) = \text{JACCARD}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



D1和D2的JACCARD相似度是?

A: 1/6

B: 1/7





# 文本表示1：TF-IDF、SVD

- 独热编码 (One Hot Encoding)

- 基于集合的文本相似度计算

- D1: "ides of March"

- A = {ides, of, March}

- D2: "Caesar died in March"

- B = {Caesar, died, in, March}

$$\text{sim}(D1, D2) = \text{JACCARD}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$= \frac{| \{\text{March}\} |}{| \{\text{ides, of, March, Caesar, died, in}\} |}$$

$$= \frac{1}{6}$$

D1和D2的JACCARD相似度是？

A: 1/6

B: 1/7



# 文本表示1: TF-IDF、SVD

- **独热编码** (One Hot Encoding)

- 优点

- 1. 简单
    - 2. 直观、容易理解
    - 3. 运算效率高

- 缺点

- 1. 实际应用中词汇表很大，每个单词都是高维、稀疏向量，后续处理困难
    - 2. 单词的向量之间，没有体现出应有的相似性(比如car和automobile)

这些问题，由后续各种表示法陆续解决

- “We are different, different, different”这句话里面，different出现3次，显得很重要。
- 在One Hot Encoding基础上，这个文档的表示，没有多个different的体现

- 未考虑单词在文本中出现的**词频**，比如 “The cat sat on the mat” v.s. “the cat sat on the mat mat”
- 未考虑单词在文本中出现的**位置**，比如 “The cat sat on the mat” v.s. “the mat sat on the cat”，以及 “hot dog” v.s. “dog hot”
- **使用频率较低的词**往往带有更大的信息量，比如 “希格斯玻色子” v.s. “好”

# 文本表示1: TF-IDF、SVD



# 文本表示1: TF-IDF、SVD

- 绝对词频

- 每一个文档表示成一个N维向量，每一维对应一个单词
  - 如果该单词出现在文档中，设置为这个词在文档中的出现的**次数（即频率）**
  - Counter = occurrence frequency**
  - 如果对应的单词未出现在文档中，设置为0

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	157	73	0	0	0	1	
BRUTUS	4	157	0	2	0	0	
CAESAR	232	227	0	2	1	0	
CALPURNIA	0	10	0	0	0	0	
CLEOPATRA	57	0	0	0	0	0	
MERCY	2	0	3	8	5	8	
WORSER	2	0	1	1	1	5	
...							



# 文本表示1: TF-IDF、SVD

- 绝对词频仍然存在问题
  - 高维**:  $N$  = 字典的大小 (例如:  $N = 100,000$ )
  - 稀疏**: 一个文档长度远小于字典大小, 仅仅出现字典中少量词, 大部分维度为0, 后续处理困难

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	157	73	0	0	0	1	
BRUTUS	4	157	0	2	0	0	
CAESAR	232	227	0	2	1	0	
CALPURNIA	0	10	0	0	0	0	
CLEOPATRA	57	0	0	0	0	0	
MERCY	2	0	3	8	5	8	
WORSER	2	0	1	1	1	5	
...							

# 文本表示1: TF-IDF、SVD

- 绝对词频

- 绝对词频有一个明显的**不合理性**

- 内容较长的文档更有可能比内容较短的文档，出现更多的关键字(Key Word)

- 虽然长文档出现更多的关键字，但是相对于整个文档长度来讲，关键字显得**相当稀疏**
      - 而短文档虽然出现更少的关键字，但是相对于整个文档长度来讲，关键字可能显得**相当密集**

杨振宁XXXXXXXXXXXXXXXXXXXXX  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XXXXXXX杨振宁XXXXXXXXXXXXX  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX



比较

杨振宁XXXXXXXXXXXXXXXXXXXXX  
XXXXXXXXXXXXX杨振宁XXXXXX

再打个比方：把两颗糖放入一大锅水，和把两颗糖放入一杯水，那个更甜？

# 文本表示1: TF-IDF、SVD

- **相对词频**TF(Term Frequency)
  - 相对词频(Term Frequency, TF)的计算方法为,
  - $TF = \frac{\text{该词项(Term)在该文档出现的次数}}{\text{该文档的词项的总数}}$
  - 这个值越大, 表示这个词项越重要

比如, 对一篇文档进行分词以后, 总共有500个词项, 词项“World”出现的次数是3次, 那么其TF值为  $TF = 3/500=0.006$



# 文本表示1: TF-IDF、SVD

- 相对词频TF(Term Frequency)

- 用TF表示一个词项还不够
- 一个词项出现的文档数越少，它越能够把文档区分出来，于是就越重要
- 或者反过来说，一个词项，如果在每篇文档里都出现，那么它就没有那么重要了

- Doc1:我们的生活好
- Doc2:我们的工作好
- Doc3:你们的生活好
- Doc4:你们的工作好
- Doc5:他们的生活好
- Doc6:他们的工作好
- Doc7:希格斯玻色子

1, 在左侧的一系列文档里, “的”、“好”几乎每个文档都出现, 没那么重要, 区分度不够  
2, “希格斯玻色子”只在少数文档出现, 所以显得很稀有, 很突兀, 也很有区分度



# 文本表示1: TF-IDF、SVD

- 逆文档频率IDF (Inverse Document Frequency)
  - 单词具有不同的重要度, **使用频率较低**的词往往带有更大的信息量
  - “希格斯玻色子” vs. “好”
  - 单词的重要性计算: **Inverse Document Frequency (IDF)**
  - $IDF(t) = \log_{10} \frac{N}{DF(t)}$

$DF(t)$ : 给定一个含有 $N$ 个文档的集合 $D$ , 统计这个单词被多少个文档使用

- Doc1:我们的生活好
- Doc2:我们的工作好
- Doc3:你们的生活好
- Doc4:你们的工作好
- Doc5:他们的生活好
- Doc6:他们的工作好
- Doc7:希格斯玻色子

1, 在左侧的一系列文档里, “的”、“好”几乎每个文档都出现, 没那么重要, 区分度不够  
2, “希格斯玻色子”只在少数文档出现, 所以显得很稀有, 很突兀, 也很有区分度

# 文本表示1: TF-IDF、SVD

- IDF的计算实例 (文档数量 $N=1,000,000$ )

Compute  $\text{idf}_t$  using the formula:  $\text{idf}_t = \log_{10} \frac{1,000,000}{\text{df}_t}$

term	$\text{df}_t$	$\text{idf}_t$
calpurnia	1	
animal	100	
sunday	1000	
fly	10,000	
under	100,000	
the	1,000,000	

基于上述公式计算左表中 $\text{df}$ 对应的 $\text{idf}$ 值

A: 6, 4, 3, 2, 1, 0

B: 0, 1, 2, 3, 4, 6

# 文本表示1: TF-IDF、SVD

- IDF的计算实例 (文档数量 $N=1,000,000$ )

Compute  $\text{idf}_t$  using the formula:  $\text{idf}_t = \log_{10} \frac{1,000,000}{\text{df}_t}$

term	$\text{df}_t$	$\text{idf}_t$
calpurnia	1	6
animal	100	4
sunday	1000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

基于上述公式计算左表中 $\text{df}$ 对应的 $\text{idf}$ 值

A: 6, 4, 3, 2, 1, 0

B: 0, 1, 2, 3, 4, 6

# 文本表示1: TF-IDF、SVD

- **TF-IDF**: 把TF和IDF结合起来
  - 单词重要性 (IDF) 与单词在文档中的频率 (TF) 结合
$$\text{tf-idf}(t, d) = \text{tf}(t, d) * \text{IDF}(t)$$

	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	
...							

- 每一个文档表示成一个N维向量, 每一维对应一个单词
  - 如果该单词出现在文档中
    - 设置为这个词在文档中的出现的**TF-IDF值 (实数值向量)**
  - 如果对应的单词未出现在文档中, 设置为0





# 文本表示1: TF-IDF、SVD

- TF-IDF: 实现文档空间→向量空间的表示

- TF-IDF表示, 将文档表示到一个向量空间

- 空间维度:  $|V|$ , 其中 $V$ 为字典集合
    - 坐标轴: 单词
    - 每一个文档表示为空间内的一个 $|V|$ 维实数值向量:  $\mathbf{d} \in R^{|V|}$

- 特点

- 高维:  $|V|$ 可能有上百万维 (集合中不同单词的数目)
    - 稀疏: 一个文档的长度有限

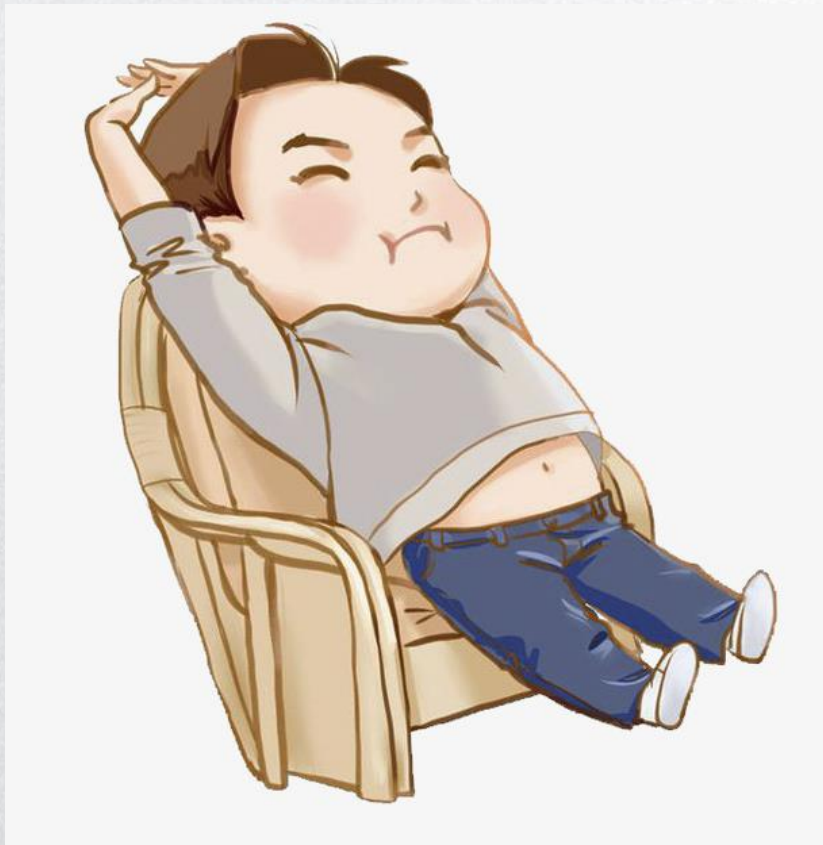
- 两个文本的相似度计算: 向量空间中可根据需要, 定义多种相似度

- 点积: dot product
    - 向量夹角余弦: cosine
    - ... ..

- 表示能力: 向量比集合能够容纳丰富信息, 如: 词频、文档频率、文档长度等
- 运算能力: 基于向量可以定义更强大的运算 (线性代数), 如: 点积、余弦.....

TF-IDF被广泛用于文档的分类、聚类、话题分析等

# 文本表示1: TF-IDF、SVD





# 文本表示1: TF-IDF、SVD

## • 分布式表示入门

绝对词频、相对词频-逆文档频率等都有此类问题

- 独热表示的不足 (1)
- 从表示语义的角度看, 高维、稀疏的独热表示还有巨大改进空间
  - 每一个单词对应一个坐标轴
  - 任意两个坐标轴垂直  $\rightarrow$  任意不同的单词语义无关
  - 但是: 自然语言中存在大量的近义词、同义词、反义词等

man	[1,0,...,0,0,...,0,0]
woman	[0,1,...,0,0,...,0,0]
car	[0,0,...,1,0,...,0,0]
automobile	[0,0,...,0,0,...,1,0]

坐标1

坐标2

$$\cos(\text{car}, \text{automobile}) = 0!$$

$$\cos(\text{man}, \text{women}) = \cos(\text{man}, \text{car})$$





# 文本表示1: TF-IDF、SVD

- 分布式表示入门
  - 独热表示的不足 (2)
  - 语义 “可泛化性”
    - 虽然训练文本中没有出现过词组 “three groups”, 但是人可以意识到 “three groups” 有很大概率是一个正确的表示, 即  $P(\text{groups}|\text{three}) > 0$
    - 独热表示没有泛化能力, 很容易导出  $P(\text{groups} | \text{three}) = 0$

Training corpus:

- ❖ There are three teams left for the qualification.
- ❖ four teams have passed the first round.
- ❖ four groups are playing in the field.





# 文本表示1: TF-IDF、SVD

- 分布式表示入门
  - 单词的分布式表示
  - 将单词 (本文) 表示为**低维、稠密**的向量
    - 每一个维度**不再具有显式的意义**
    - **单词**表示为**隐空间(latent space)**中的一个向量

如何得到这些向量的具体做法，将在后续介绍

Vector Space Representation	
man	[0.326172, ..., 0.00524902, ..., 0.0209961]
woman	[0.243164, ..., -0.205078, ..., -0.0294189]
car	[0.0512695, ..., -0.306641, ..., 0.222656]
automobile	[0.107422, ..., -0.0375977, ..., -0.0620117]

Vectors from GoogleNews-vectors-negative300.bin

# 文本表示1: TF-IDF、SVD

- 分布式表示入门
  - 单词的分布式表示
  - 分布式表示的优势 (1)
    - 降低表示维度, 更少的参数和存储空间
    - 能够**承载语义信息**和运算

Distributed Representation	
man	[0.326172, ..., 0.00524902, ..., 0.0209961]
woman	[0.243164, ..., -0.205078, ..., -0.0294189]
car	[0.0512695, ..., -0.306641, ..., 0.222656]
automobile	[0.107422, ..., -0.0375977, ..., -0.0620117]

$$\cos(\text{man}, \text{women}) = 0.77$$

$$\cos(\text{man}, \text{automobile}) = 0.25$$

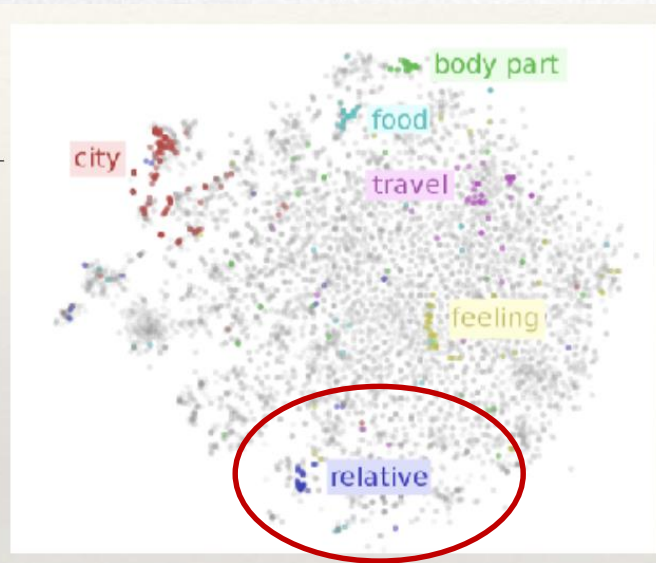
# 文本表示1: TF-IDF、SVD

- 分布式表示入门
  - 单词的分布式表示
  - 基于分布式表示的相似单词
    - 对单词的向量表示进行降维, 可视化, 发现相似的单词聚集在一块

FRANCE 454	JESUS 1973	XBOX 6909	REDDISH 11724	SCRATCHED 29869	MEGABITS 87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

What words have embeddings closest to a given word?

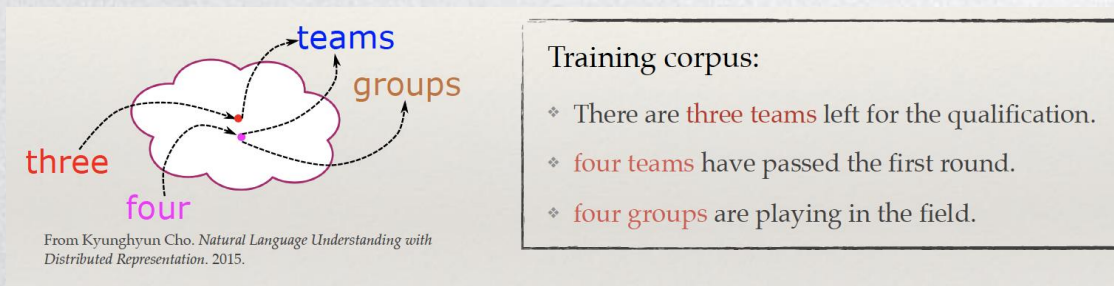
From Collobert et al. (2011)





# 文本表示1: TF-IDF、SVD

- 分布式表示入门
  - 单词的分布式表示
  - 分布式表示的优势 (2)
    - 具有 “语义可泛化性”
    - 单词 “three” 和 “four” 在隐空间的表示相似
    - 单词 “teams” 和 “groups” 在隐空间内的表示相似
    - → 训练数据中未出现的bigram: three groups



$$P(\text{groups} \mid \text{four}) > 0, P(\text{teams} \mid \text{three}) > 0 \rightarrow P(\text{groups} \mid \text{three}) > 0$$





# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现

- 先来猜一下单词 “bardiwac” 大致的含义

- ▶ He handed her her glass of bardiwac.
- ▶ Beef dishes are made to complement the bardiwacs.
- ▶ Nigel staggered to his feet, face flushed from too much bardiwac.
- ▶ Malbec, one of the lesser-known bardiwac grapes, responds well to Australia's sunshine.
- ▶ I dined off bread and cheese and this excellent bardiwac.
- ▶ The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

A: 一种红葡萄酒

B: 一个地名

C: 一个人名



# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现

- 先来猜一下单词 “bardiwac” 大致的含义

- ▶ He handed her her glass of bardiwac.
- ▶ Beef dishes are made to complement the bardiwacs.
- ▶ Nigel staggered to his feet, face flushed from too much bardiwac.
- ▶ Malbec, one of the lesser-known bardiwac grapes, responds well to Australia's sunshine.
- ▶ I dined off bread and cheese and this excellent bardiwac.
- ▶ The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

A: 一种红葡萄酒

B: 一个地名

C: 一个人名

# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现
  - 为什么可以猜到?

Bardiwac在多个文档中的上下文是关键!!!

the doctor. `</p><p>` `Just checking on the **bardiwac** , he boomed as he came back. `Edith's very  
`</p><p>` `I hope you'll take to a good French **bardiwac** , ' chimed in Arthur Iverson jovially. `One  
`Our host did slip out to attend to the **bardiwac** &hellip;' `</p><p>` `That was before the shrimp  
Iverson did when he went through to see to the **bardiwac** before dinner.' Henry rubbed his hands.  
and drinking red wine from France -- sour **bardiwac** , which had proved hard to sell. The room  
eyes were alight and he was drinking the **bardiwac** down like water. `It is like Hallow-fair  
quizzically at him and offering him some more **bardiwac** . `</p><p>` He shook his head. `I will sleep  
drinks (as Queen Victoria reputedly did with **bardiwac** and malt whisky), but still the result  
Do we really `wash down' a good meal with **bardiwac** ? Port is immediately suggested by Stilton  
completely different: cheap and cheerful **bardiwac** . Two good examples from Victoria Wine are  
examples from Victoria Wine are its house **bardiwac** , juicy and a touch almondy, a good buy  
opened a bottle of rather rust-coloured **bardiwac** . I ate too much and drank nearly three-quarters  
elections, it was apparent the SDP of ` **bardiwac** and chips' mould-breaking fame at the time  
the black hills. Not a night of vintage **bardiwac** . `</p><p>` Burnley: Pearce, Measham, McGrory  
SONS Old School -- the Marlborian navy, **bardiwac** and slim-white stripe. Heavy woven silk  
white-hot passion. We are like a good bottle of **bardiwac** ; we both have sediment in our shoes. `</p>`  
few minutes later he was uncorking a fine **bardiwac** in Masha's room, saying he had something  
the phone. Surkov silently offered me more **bardiwac** but I indicated a bottle of Perrier. `</p>`  
defenders as Villa swept past them like a **bardiwac** and blue tidal wave. `</p><p>` Things are difficult  
campaign. Refreshed by a nimble in-flight **bardiwac** , they serenaded him with a special song





# 文本表示1：TF-IDF、SVD

- 分布式表示入门：如何实现
  - 语言学中的分布式假设

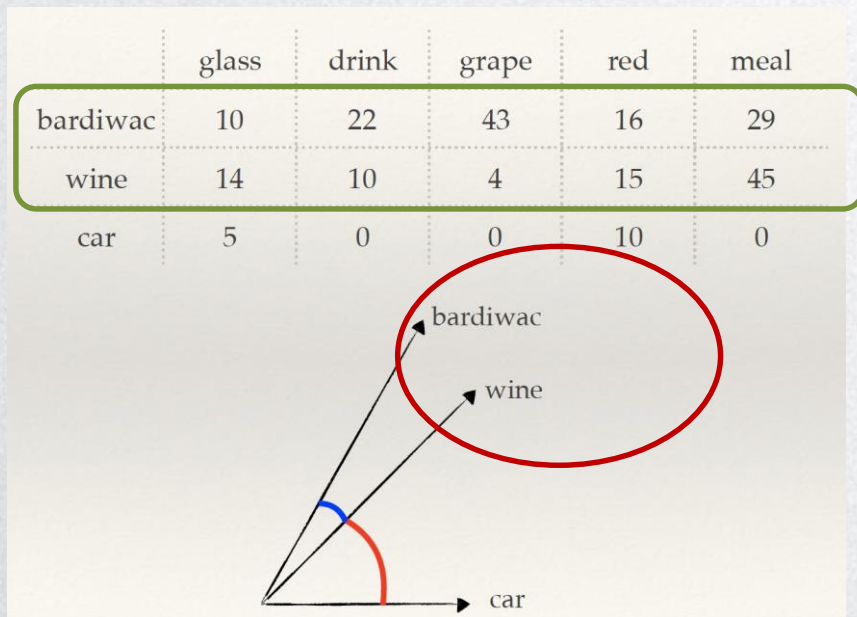
“The meaning of a word lies in its use.”

- ▶ “Die Bedeutung eines Wortes liegt in seinem Gebrauch.”  
— Ludwig Wittgenstein
- ▶ “You shall know a word by the company it keeps!”  
— J. R. Firth (1957)
- ▶ Distributional hypothesis (Zellig Harris 1954)



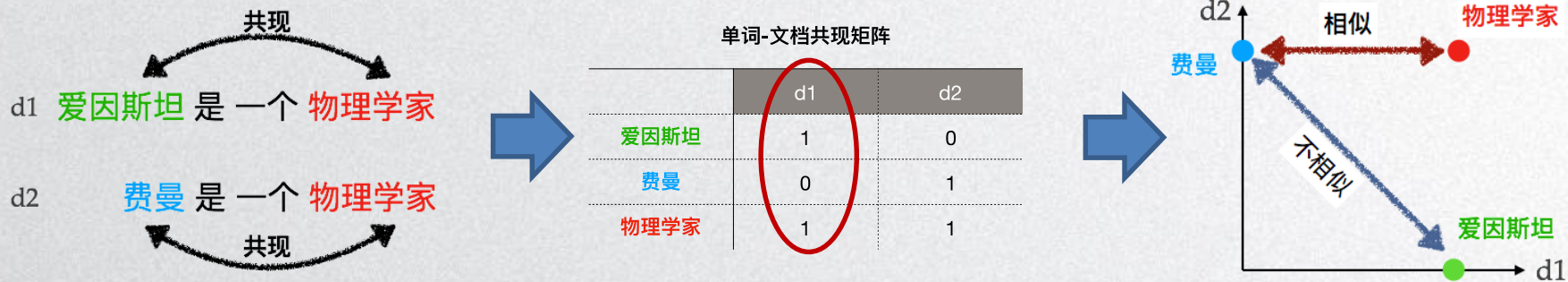
# 文本表示1: TF-IDF、SVD

- 分布式表示入门：如何实现
  - 单词的语义隐藏在它们的共现数据中



# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现
  - 分布式假设的表现形式1 (文档内共现)



适合长文本的表达: 单词**共现**关系建模

话题模型: LSI、NMF、PLSI、LDA等

# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现
  - 分布式假设的表现形式2 (上下文共现)



适合短文本的表达: 单词**聚合**关系建模

词嵌入模型: Word2Vec, Glove, BTM等词嵌入模型



# 文本表示1: TF-IDF、SVD

- 分布式表示入门: 如何实现
  - 隐空间中与Feynman相近的词

spacetime tachyons  
quantum physicist  
**feynman**  
geometrodynamics electrodynamics  
schwinger gravitation

基于**话题模型**得到的语义相近的词

heisenberg bethe  
schwinger einstein  
**feynman**  
schrodinger bohr  
hamiltonian

基于**词嵌入模型**关系得到的语义相近的词



# 文本表示1: TF-IDF、SVD





# 文本表示1：TF-IDF、SVD

- 话题模型LSI——**潜语义索引**(Latent Semantic Index, LSI)
  - 也被称为潜在语义分析(Latent Semantic Analysis, LSA)
  - 基于**矩阵分解**的话题模型
  - 通过求解单词-文档共现矩阵的**低秩近似**得到话题

lsa.colorado.edu › papers › JASIS.lsi.90.pdf ▾ PDF

## Indexing by Latent Semantic Analysis Scott Deerwester ...

The particular "**latent semantic indexing**" (LSI) analysis that we have tried uses ... The Voorhees data were obtained directly from her **paper** in which she used.

by S Deerwester - **Cited by 14885** - Related articles

Deerwester et al., Indexing by Latent Semantic Analysis. 1990.

# 文本表示1：TF-IDF、SVD

- 潜在语义索引(Latent Semantic Index, LSI)
- 通过求解单词-文档共现矩阵的**低秩近似**得到话题
  - 目标：去除掉文档共现矩阵中的“噪音”
  - 解决思路：找到一个秩为k的矩阵，使其与文档近似矩阵最为相近。

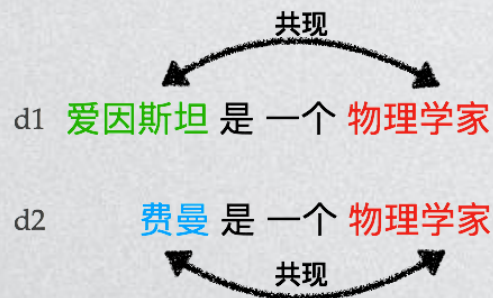
$$\begin{aligned}\hat{Z} &= \operatorname{argmin}_{Z | \operatorname{rank}(Z)=k} \|C - Z\|_F \\ &= \operatorname{argmin}_{Z | \operatorname{rank}(Z)=k} \sqrt{\sum_{i=1}^M \sum_{j=1}^N (C_{ij} - Z_{ij})^2}\end{aligned}$$

- LSI的原理：发现词项-文档矩阵中的**低秩结构**
- **分布式语义假设**：语义相关的词倾向共现在同一个文档中



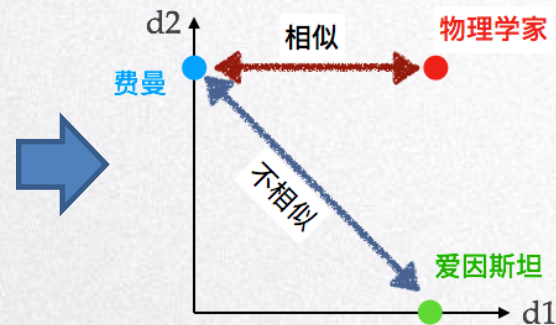
# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始



单词-文档共现矩阵

	d1	d2
爱因斯坦	1	0
费曼	0	1
物理学家	1	1





# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始

单词-文档共现矩阵可以为:

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	157	73	0	0	0	1	
BRUTUS	4	157	0	0	0	0	
CAESAR	232	227	0	0	1	0	
CALPURNIA	0	10	0	0	0	0	
CLEOPATRA	57	0	0	0	0	0	
MERCY	2	0	3	0	5	8	
WORSER	2	0	1	0	1	5	
...							

词频

或者

	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	0.0	0.0	0.0	
CAESAR	8.59	2.54	0.0	0.0	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	0.0	5.25	0.88	
WORSER	1.37	0.0	0.11	0.0	0.25	1.95	
...							

TF-IDF

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - LSI依赖于奇异值分解(Singular Value Decomposition, SVD)

举个例子

- 1, 了解如何进行奇异值分解
- 2, 以及分解以后各个矩阵的意义

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - LSI依赖于奇异值分解(Singular Value Decomposition, SVD)
  - 假设有矩阵A如下

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\text{计算 } AA^T = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

对 $AA^T$ 进行特征分解, 有

$$\lambda_1 = 3; u_1 = \begin{pmatrix} 1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{pmatrix}; \lambda_2 = 1; u_2 = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{pmatrix};$$

$$\lambda_3 = 0; u_3 = \begin{pmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix}$$

构造U

$$\begin{pmatrix} 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \\ 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \end{pmatrix}$$

U的各个  
列正交

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - LSI依赖于奇异值分解(Singular Value Decomposition, SVD)
  - 假设有矩阵A如下

$$- A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$- \text{计算 } A^T A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$- = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

对 $A^T A$ 进行特征分解, 有

$$\lambda_1 = 3; v_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}; \lambda_2 = 1; v_2 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix};$$

构造 $v$

$$\begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

$v$ 的各个  
列正交



# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - LSI依赖于奇异值分解(Singular Value Decomposition, SVD)
  - 假设有矩阵A如下

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\text{计算 } A^T A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

根据特征值

$$\lambda_1 = 3; \lambda_2 = 1; \lambda_3 = 0;$$

构造准对角矩阵 $\Sigma$

$$\begin{pmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{pmatrix}$$

奇异值为特征值的开根方

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - LSI依赖于奇异值分解(Singular Value Decomposition, SVD)
  - 于是有

$$A = U \sum V^T$$

注意转置

$$= \begin{pmatrix} 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \\ 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

正交

正交

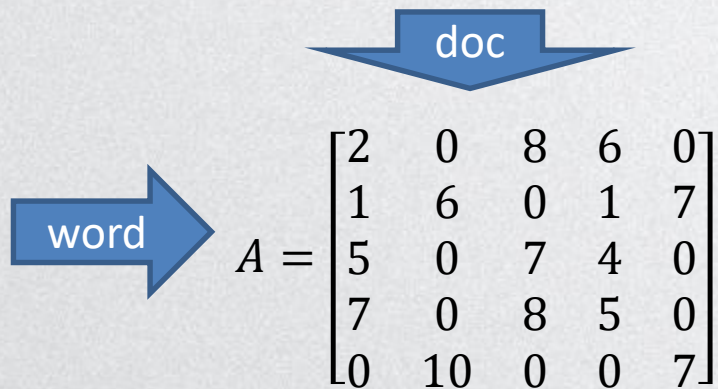
# 文本表示1: TF-IDF、SVD





# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 从单词-文档的共现矩阵开始
  - 假设有词项-文档矩阵A
  - 对矩阵A进行奇异值分解, 有  $A = U S V^T$


$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix}$$

如何进行矩阵的奇异值分解SVD, 前文已经介绍



# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)

- 对矩阵A进行奇异值分解, 有  $A = U S V^T$

doc

word

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix} = U S V^T$$

SVD分解后,  $U S V^T$   
可以重建A

$$U = \begin{bmatrix} -0.54 & 0.065 & 0.82 & 0.11 & -0.12 \\ -0.10 & -0.59 & -0.11 & 0.79 & 0.06 \\ -0.525 & 0.06 & -0.21 & -0.12 & 0.81 \\ -0.645 & 0.07 & -0.51 & -0.06 & -0.56 \\ -0.06 & -0.80 & 0.09 & -0.59 & -0.04 \end{bmatrix}$$

$$S = \begin{bmatrix} 17.92 & 0 & 0 & 0 & 0 \\ 0 & 15.17 & 0 & 0 & 0 \\ 0 & 0 & 3.564 & 0 & 0 \\ 0 & 0 & 0 & 1.984 & 0 \\ 0 & 0 & 0 & 0 & 0.3496 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.465 & -0.07 & -0.735 & -0.484 & -0.065 \\ 0.022 & -0.76 & 0.099 & 0.025 & -0.64 \\ -0.869 & 0.063 & 0.28 & 0.399 & -0.0442 \\ 0.0008 & -0.60 & -0.223 & 0.33 & 0.70 \\ -0.17 & -0.228 & 0.565 & -0.704 & 0.323 \end{bmatrix}$$

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)

- 对矩阵A进行奇异值分解, 有  $A = U S V^T$

word →

doc ↓

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix} = U S V^T$$

A为词项文档-矩阵,  $M \times N$ 的矩阵

- 那么U的每一行表示一个单词
- 每一列代表一个概念, 绝对值越大表示相关性越高; 比如单词1和概念1、3的相关性较高

U: 左特征向量

- U的第1列, 对应由所有**单词**张成的空间中, 方差最大的方向
- U的第2列, 对应与U的第1列垂直的所有方向中, 方差最大的方向

.....

$$U = \begin{bmatrix} -0.54 & 0.065 & 0.82 & 0.11 & -0.12 \\ -0.10 & -0.59 & -0.11 & 0.79 & 0.06 \\ -0.525 & 0.06 & -0.21 & -0.12 & 0.81 \\ -0.645 & 0.07 & -0.51 & -0.06 & -0.56 \\ -0.06 & -0.80 & 0.09 & -0.59 & -0.04 \end{bmatrix}$$

$$S = \begin{bmatrix} 17.92 & 0 & 0 & 0 & 0 \\ 0 & 15.17 & 0 & 0 & 0 \\ 0 & 0 & 3.564 & 0 & 0 \\ 0 & 0 & 0 & 1.984 & 0 \\ 0 & 0 & 0 & 0 & 0.3496 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.465 & -0.07 & -0.735 & -0.484 & -0.065 \\ 0.022 & -0.76 & 0.099 & 0.025 & -0.64 \\ -0.869 & 0.063 & 0.28 & 0.399 & -0.0442 \\ 0.0008 & -0.60 & -0.223 & 0.33 & 0.70 \\ -0.17 & -0.228 & 0.565 & -0.704 & 0.323 \end{bmatrix}$$

# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)

- 对矩阵A进行奇异值分解, 有  $A = U S V^T$

doc

word

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix} = U S V^T$$

A为词项文档-矩阵

1, 那么  $V^T$  的每一列表示一个文档, 每一行代表一个概念, 绝对值越大表示相关性越高; 比如文档2和概念2、4的相关性较高

V: 右特征向量

1, V的第1列, 对应由所有文档张成的空间中, 方差最大的方向  
2, V的第2列, 对应与V的第1列垂直的所有方向中, 方差最大的方向.....注意右侧的矩阵已经转置过了

$$U = \begin{bmatrix} -0.54 & 0.065 & 0.82 & 0.11 & -0.12 \\ -0.10 & -0.59 & -0.11 & 0.79 & 0.06 \\ -0.525 & 0.06 & -0.21 & -0.12 & 0.81 \\ -0.645 & 0.07 & -0.51 & -0.06 & -0.56 \\ -0.06 & -0.80 & 0.09 & -0.59 & -0.04 \end{bmatrix}$$

$$S = \begin{bmatrix} 17.92 & 0 & 0 & 0 & 0 \\ 0 & 15.17 & 0 & 0 & 0 \\ 0 & 0 & 3.564 & 0 & 0 \\ 0 & 0 & 0 & 1.984 & 0 \\ 0 & 0 & 0 & 0 & 0.3496 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.465 & -0.07 & -0.735 & -0.484 & -0.065 \\ 0.022 & -0.76 & 0.099 & 0.025 & -0.64 \\ -0.869 & 0.063 & 0.28 & 0.399 & -0.0442 \\ 0.0008 & -0.60 & -0.223 & 0.33 & 0.70 \\ -0.17 & -0.228 & 0.565 & -0.704 & 0.323 \end{bmatrix}$$



# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)

- 对矩阵A进行奇异值分解, 有  $A = U S V^T$

doc

word

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix} = U S V^T$$

A为词项文档-矩阵

1, S矩阵对角线上的值, 可以理解为5个概念的强度

.....注意, 对角线上的值, 从左上角到右下角是从大到小排序的

$$U = \begin{bmatrix} -0.54 & 0.065 & 0.82 & 0.11 & -0.12 \\ -0.10 & -0.59 & -0.11 & 0.79 & 0.06 \\ -0.525 & 0.06 & -0.21 & -0.12 & 0.81 \\ -0.645 & 0.07 & -0.51 & -0.06 & -0.56 \\ -0.06 & -0.80 & 0.09 & -0.59 & -0.04 \end{bmatrix}$$

$$S = \begin{bmatrix} 17.92 & 0 & 0 & 0 & 0 \\ 0 & 15.17 & 0 & 0 & 0 \\ 0 & 0 & 3.564 & 0 & 0 \\ 0 & 0 & 0 & 1.984 & 0 \\ 0 & 0 & 0 & 0 & 0.3496 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.465 & -0.07 & -0.735 & -0.484 & -0.065 \\ 0.022 & -0.76 & 0.099 & 0.025 & -0.64 \\ -0.869 & 0.063 & 0.28 & 0.399 & -0.0442 \\ 0.0008 & -0.60 & -0.223 & 0.33 & 0.70 \\ -0.17 & -0.228 & 0.565 & -0.704 & 0.323 \end{bmatrix}$$



# 文本表示1: TF-IDF、SVD

- 潜语义索引(Latent Semantic Index, LSI)
  - 对矩阵A进行奇异值分解, 有  $A = U S V^T$
  - 我们对数据进行降维, 取  $K=3$ 
    - U保留前3列为  $U'$ , S保留前3行前3列为  $S'$ ,  $V^T$ 保留前3行为  $V'^T$
    - $U'$ 的每一行代表一篇文档, 每一列代表降维至3维后每篇文档在各个维度方向的投影
    - $V'^T$ 也可以类似理解
  - 对降维以后的矩阵进行重建, 即计算  $U'S'V'^T$ , 得到  $\hat{A}$ 
    - $\hat{A}$ 和A有差别; 但是差别是可以忍受的
    - 我们获得的好处, 是可以用3维向量表示每个词项和每个文档了

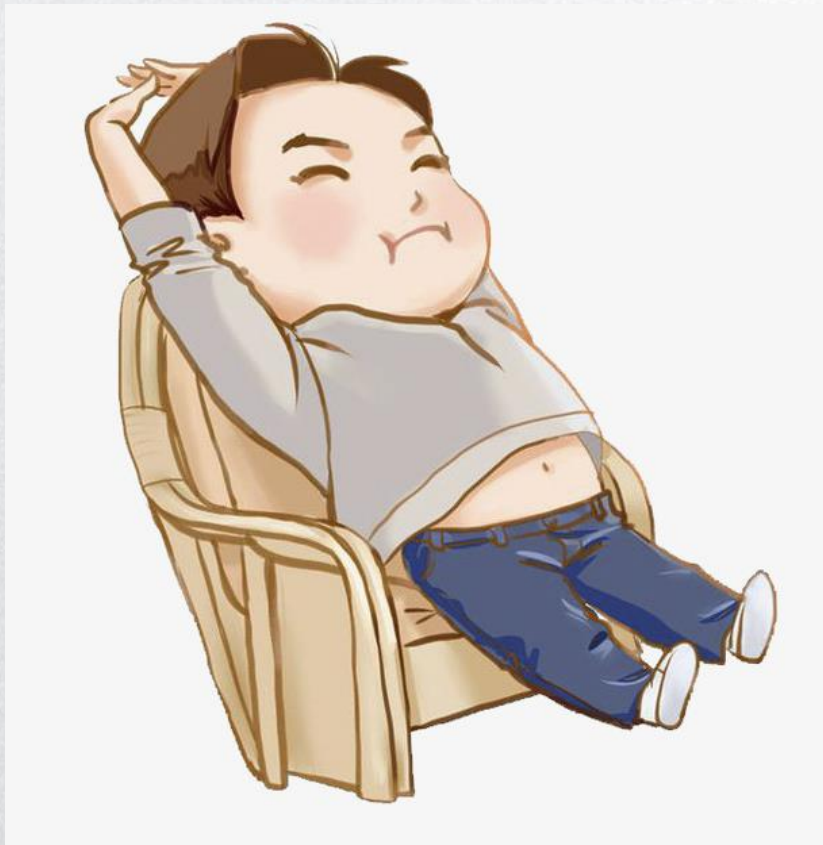
$$U'S'V'^T = \begin{bmatrix} -0.54 & 0.065 & 0.82 & 0.11 & -0.12 \\ -0.10 & -0.59 & -0.11 & 0.79 & 0.06 \\ -0.525 & 0.06 & -0.21 & -0.12 & 0.81 \\ -0.645 & 0.07 & -0.51 & -0.06 & -0.56 \\ -0.06 & -0.80 & 0.09 & -0.59 & -0.04 \end{bmatrix}$$

$$= \begin{bmatrix} 17.92 & 0 & 0 & 0 & 0 \\ 0 & 15.17 & 0 & 0 & 0 \\ 0 & 0 & 3.564 & 0 & 0 \\ 0 & 0 & 0 & 1.984 & 0 \\ 0 & 0 & 0 & 0 & 0.3496 \end{bmatrix} \begin{bmatrix} -0.465 & -0.07 & -0.735 & -0.484 & -0.065 \\ 0.022 & -0.76 & 0.099 & 0.025 & -0.64 \\ -0.869 & 0.063 & 0.28 & 0.399 & -0.0442 \\ 0.0008 & -0.60 & -0.223 & 0.33 & 0.70 \\ -0.17 & -0.228 & 0.565 & -0.704 & 0.323 \end{bmatrix}$$

$$= \begin{bmatrix} 1.98 & 0.122 & 8.04 & 5.87 & -0.133 \\ 0.991 & 6.90 & 0.318 & 0.479 & 5.88 \\ 5.05 & -0.074 & 6.79 & 4.27 & 0.0666 \\ 6.974 & -0.112 & 8.085 & 4.897 & 0.1497 \\ -0.00122 & 9.29 & -0.254 & 0.38 & 7.823 \end{bmatrix} = \hat{A} \approx A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix}$$

降维表示重建  $\leftrightarrow$  原矩阵

# 文本表示1: TF-IDF、SVD



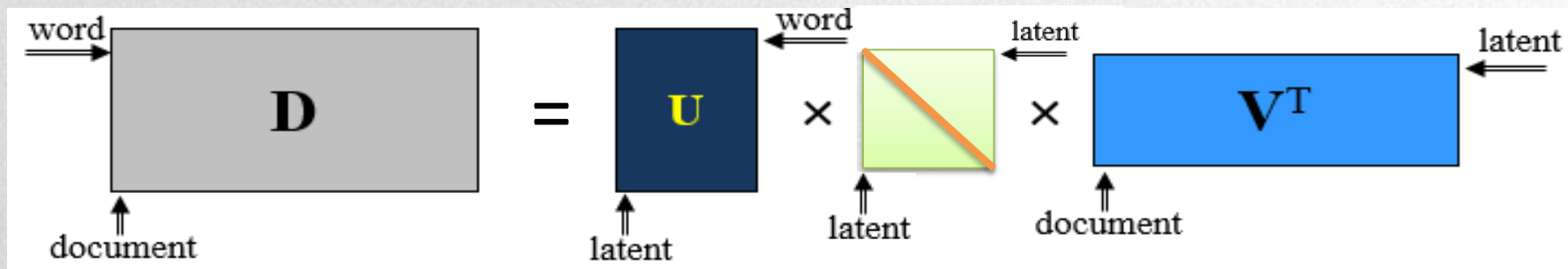
# 文本表示1：TF-IDF、SVD

- LSI第一步：SVD得到的U、 $\Sigma$ 、V

- U所有的列都正交:  $\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ ,  $\mathbf{u}_i$ 是 $\sigma_i$ 对应左特征向量

- V所有的列都正交:  $\mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ ,  $\mathbf{v}_i$ 是 $\sigma_i$ 对应右特征向量

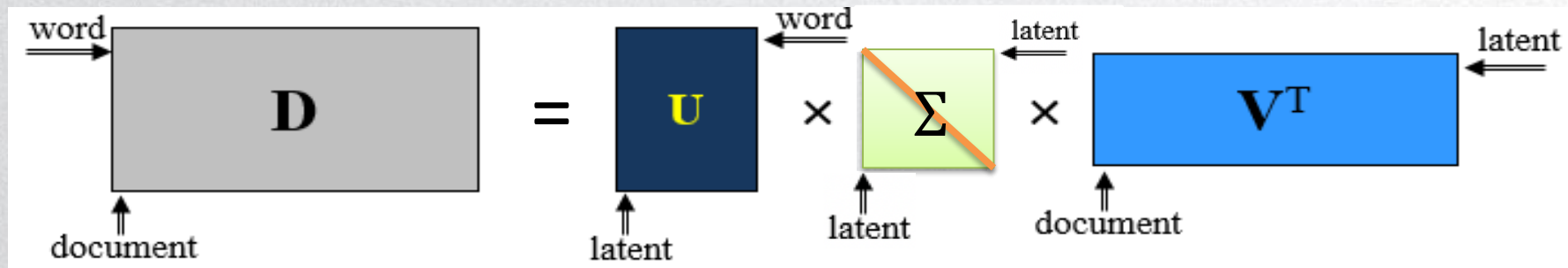
- $\Sigma = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_R \end{pmatrix}$ 是对角方阵,  $\sigma_1 \cdots \sigma_R$ 是D的R个特征根 (注意是特征值的开根方)



# 文本表示1：TF-IDF、SVD

## • LSI第二步：对特征根进行排序

- 原特征根矩阵  $\Sigma \rightarrow$  排序后的特征根矩阵  $\Sigma'$ ,  $\Sigma' = \begin{pmatrix} \sigma'_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma'_R \end{pmatrix}$ , 其中  $\sigma'_1 \geq \sigma'_2 \geq \cdots \geq \sigma'_R$
- 左/右特征向量矩阵也按照  $\Sigma'$  进行相应的调整
  - $\sigma'_i$  所对应的左特征向量调整到  $\mathbf{U}$  的第  $i$  列
  - $\sigma'_i$  所对应的右特征向量调整到  $\mathbf{V}$  的第  $i$  列





# 文本表示1: TF-IDF、SVD

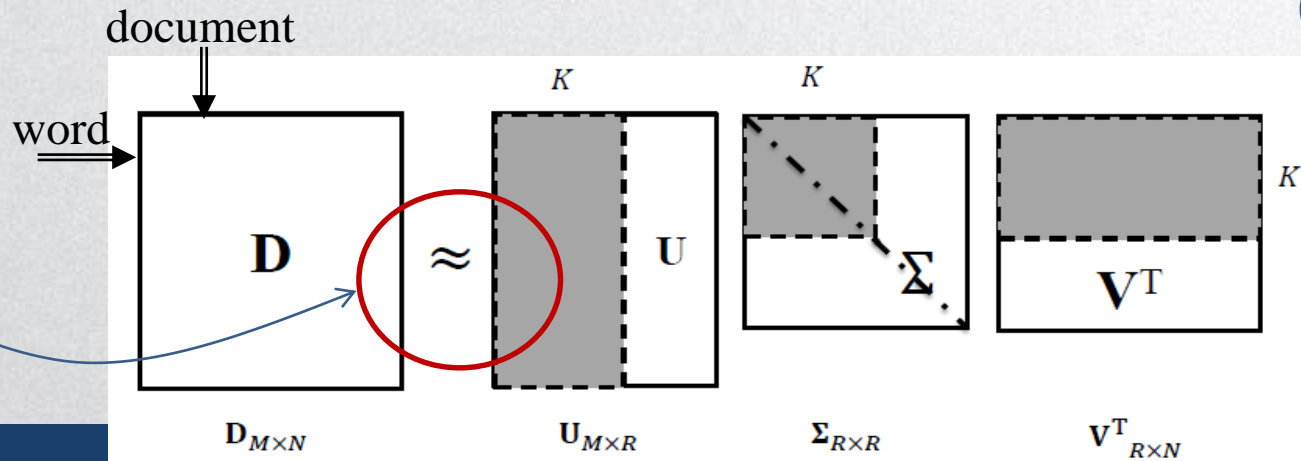
- LSI第三步: 只保留前K个特征值, 其余置零

- $\Sigma_K$ : 只保留了前K个特征根的对角阵
- 等号不再成立

K: 隐空间(latent space)的维度

- $U\Sigma_K V^T$ 是D的低秩近似

- $\text{rank}(U\Sigma_K V^T) \leq \min(\text{rank}(U), \text{rank}(\Sigma_K), \text{rank}(V)) \leq K$ 
  - 与文档/单词的数目而言, K比较小, 如: K=50、100、200、500等

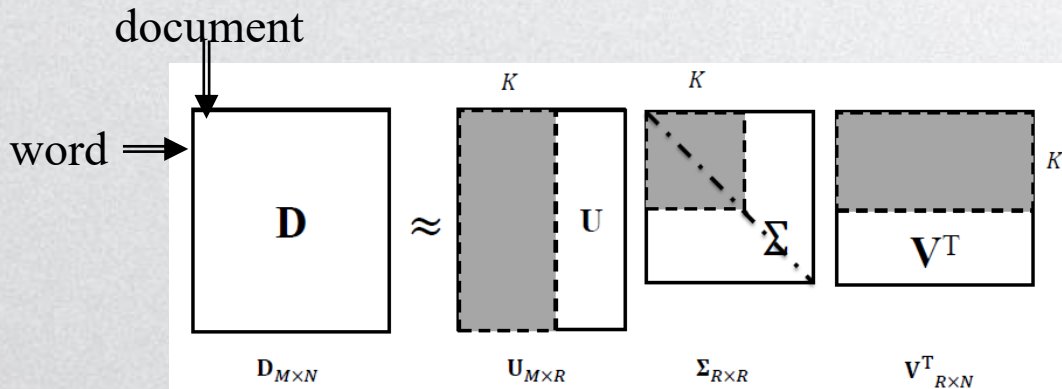


实现降维

# 文本表示1: TF-IDF、SVD

## • LSI算法总结

- 给定一个文档集合, 构建单词-文档共现矩阵 $D$  (通常为稀疏矩阵)
- 对 $D$ 进行SVD分解, 即 $D = U S V^T$
- 将特征值矩阵 $\Sigma$ 按照从大到小排序
- 保留 $K$ 个最大的特征根, 其余置0, 得到 $\Sigma_K$
- 结果
  - $U$ 矩阵的每一行, 为对应单词的表达向量( $K$ 维)
  - $V^T$ 矩阵的每一列, 为对应文档的表达向量( $K$ 维)



# 文本表示1: TF-IDF、SVD





# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
  - 有如下一系列文档, 可以认为是两个话题
  - 两类文档Title: 计算机 (c) 和数学 (m)

## Technical Memo Titles

c1: *Human machine interface for 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, ordered 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*



# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
  - 词项-文档矩阵

document  
↓  
**D**

	c1	c2	c3	c4	c5	m1	m2	m3	m4
word → 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

词项-文档矩阵：值为**单词频率**，大部分值为0

# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
  - SVD分解之U矩阵

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

latent

word  $\Rightarrow$

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

$U =$

SVD得到的U矩阵，  
为稠密矩阵（左  
特征向量的顺序  
已经调整完毕）

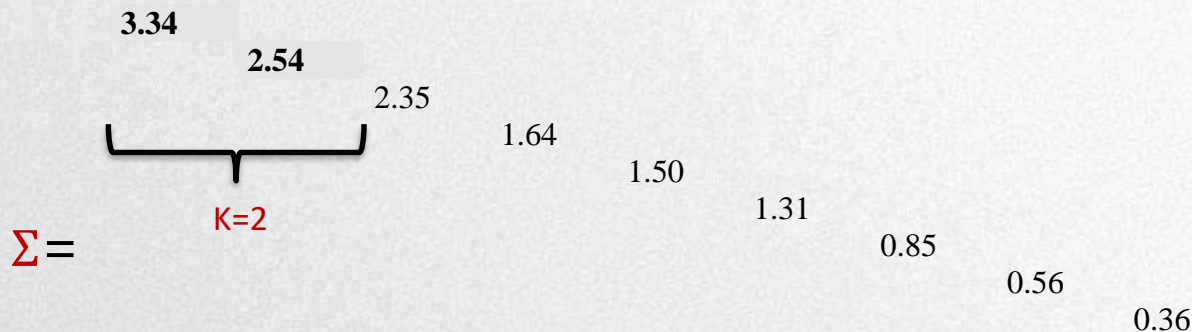
$K=2$

# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
- SVD分解之S矩阵

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

SVD得到的 $\Sigma$ 矩阵,  
为对角方阵 (特  
征根的顺序已经  
排序完毕)



# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
- SVD分解之V矩阵

SVD得到的V矩阵，为稠密阵（右特征向量的顺序已经排序完毕）

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

latent  
↓

document ⇒

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

V =

K=2



# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
  - 当K=2重建的矩阵

低秩近似得到的恢复矩阵 (K=2)，变成了稠密矩阵，即每个元素都是非0值

$$U\Sigma_K V^T =$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

# 文本表示1: TF-IDF、SVD

- 理解LSI的话题建模能力
  - 计算文档之间的文档相似度
    - 向量表示的向量间的夹角Cosine值

基于原始表达

	c1	c2	c3	c4	c5	m1	m2	m3	m4
c1	1								
c2	-0.19	1							
c3	0.00	0.00	1						
c4	0.00	0.00	0.47	1					
c5	-0.33	0.58	0.00	-0.31	1				
m1	-0.17	-0.30	-0.21	-0.16	-0.17	1			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67	1		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	1	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56	1

基于LSI

c1	c2	c3	c4	c5	m1	m2	m3	m4
1	0.91	1						
0.91	1							
1.00	0.91	1						
1.00	0.88	1.00	1					
0.85	0.99	0.85	0.81	1				
-0.85	-0.56	-0.85	-0.88	-0.45	1			
-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1		
-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	1	
-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00	1

话题  
1

话题  
2

# 文本表示1: TF-IDF、SVD

## 理解LSI的话题建模

### 可视化效果

- 两个类簇
- 两个话题

2-D Plot of Terms and Docs from Example

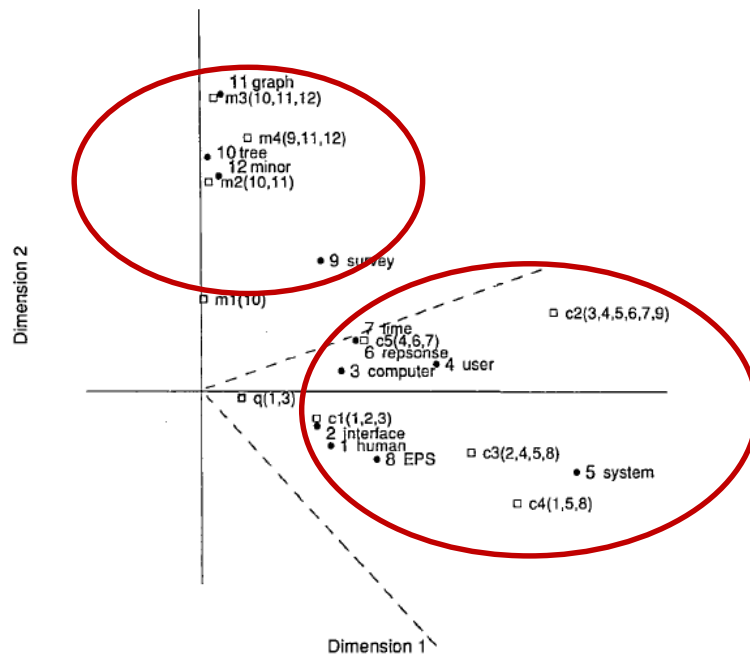


FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the same TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point  $q$ . Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query  $q$ . All documents about human-computer (c1-c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1-m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.



# 文本表示1: TF-IDF、SVD







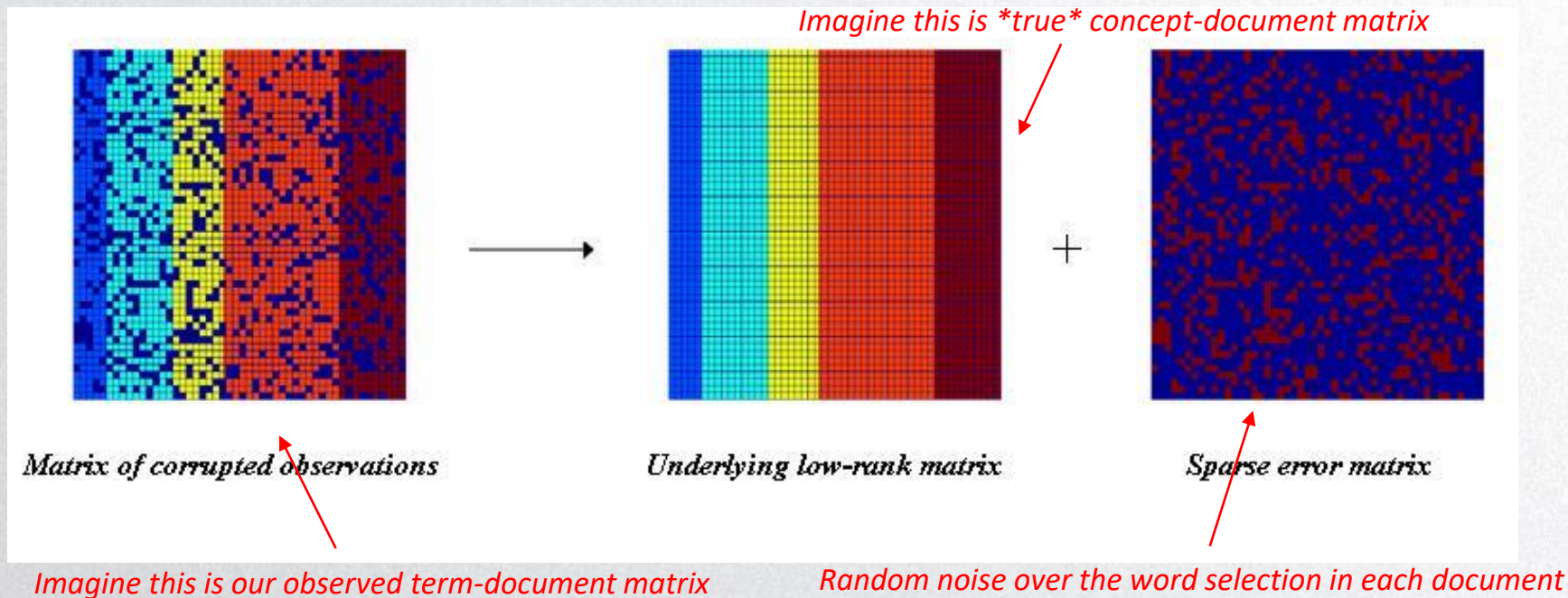
# 文本表示1: TF-IDF、SVD

- LSI总结

- 话题模型(可能是第一个?), 求解单词和文档的**分布式表达**
  - 对单词-文档矩阵进行SVD分解, 保留最大的K个特征值, 其余置0
- 是对原始输入矩阵D的低秩近似
  - 优点
    - **数学优美**
    - 有效 (被广泛应用于各种文本分析应用中)
  - 缺点
    - 低秩近似矩阵和U、V中有**负值**, 不好解释
    - **特征值、特征向量计算复杂度高**
    - M x N矩阵,  $N < M$ , 复杂度  $O(MN^2)$

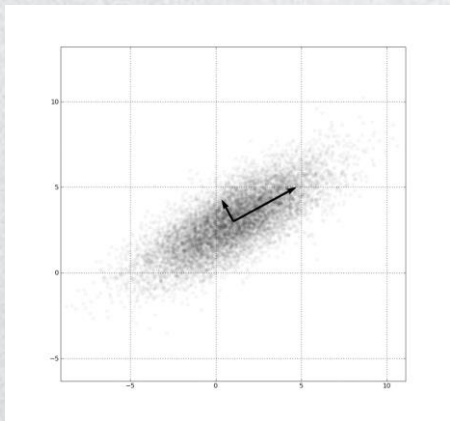
# 文本表示1: TF-IDF、SVD

- LSI总结
  - 通过求解单词-文档共现矩阵的**低秩近似**得到话题

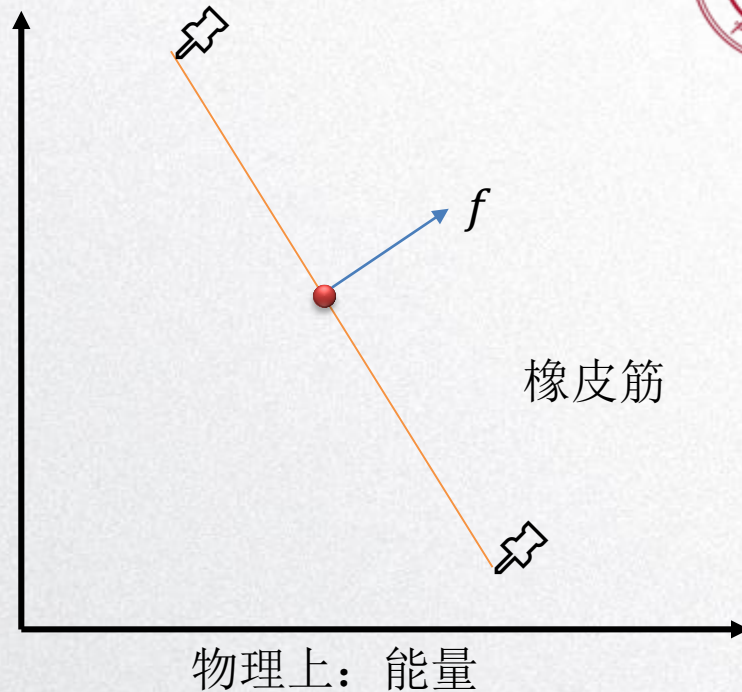


# 文本表示1: TF-IDF、SVD

- 理解LSI: 为何舍弃小的特征根?



统计上: 方差

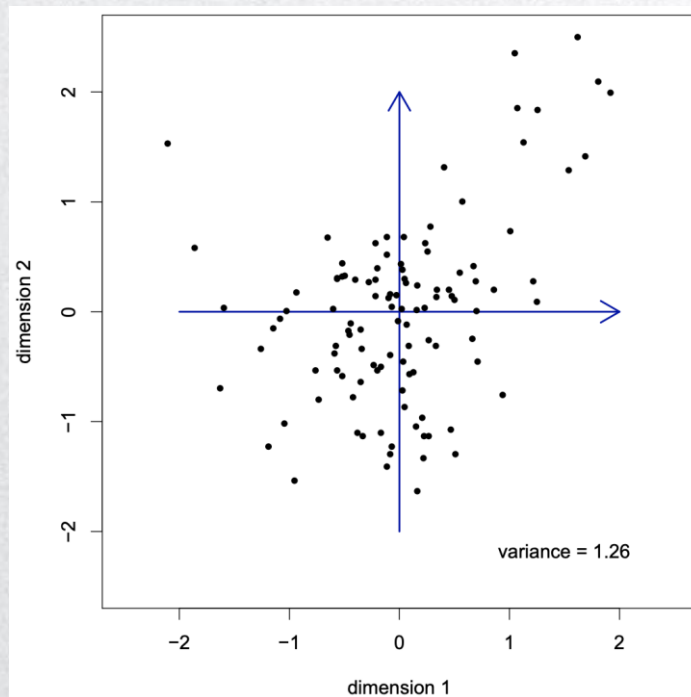


- 特征根大 $\Leftrightarrow$ 方差大 $\Leftrightarrow$ 能量大: 此方向 (特征向量方向) 承载较多有效信息
  - 特征根小 $\Leftrightarrow$ 方差小 $\Leftrightarrow$ 能量小: 此方向的波动可能是噪声
- 舍弃小的特征根
  - 降低表达维度 + 去除噪声信号



# 文本表示1: TF-IDF、SVD

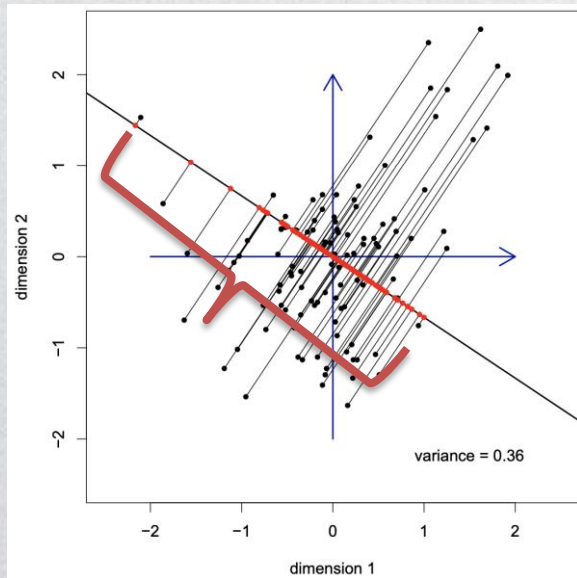
- 特征向量的方向?



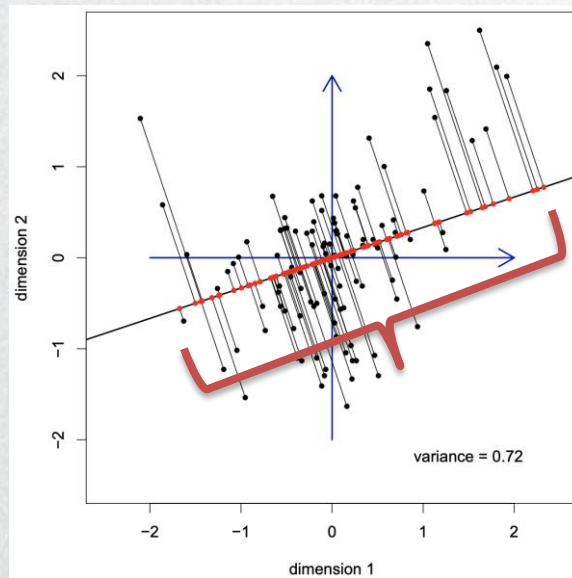


# 文本表示1: TF-IDF、SVD

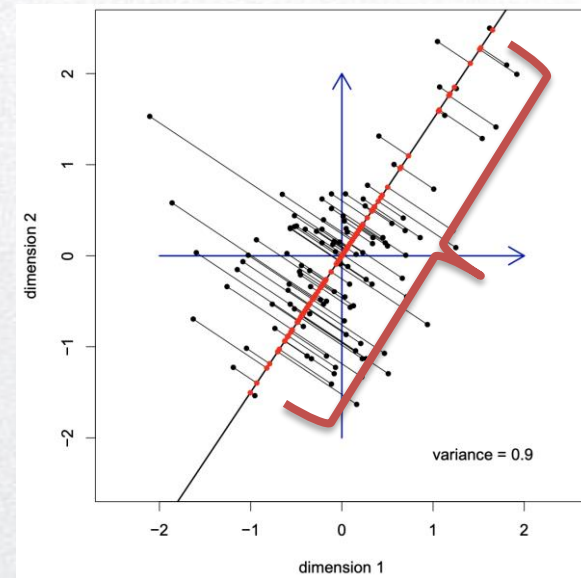
- 特征向量: 选一个方差最大的方向



方向1



方向2



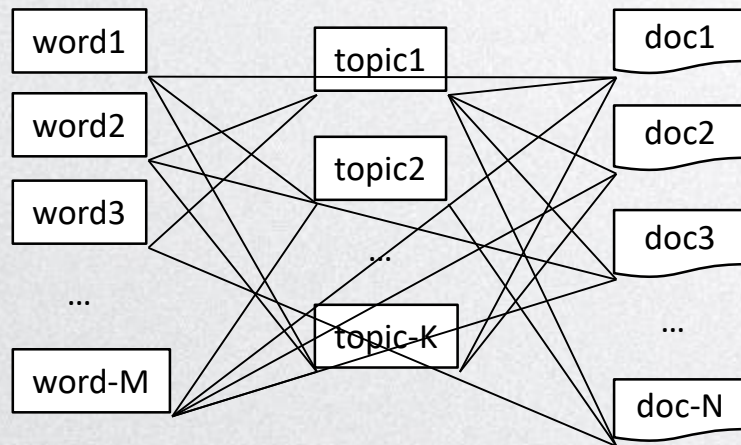
方向3

# 文本表示1: TF-IDF、SVD



# 文本表示1: TF-IDF、SVD

- 矩阵分解应用:主题模型

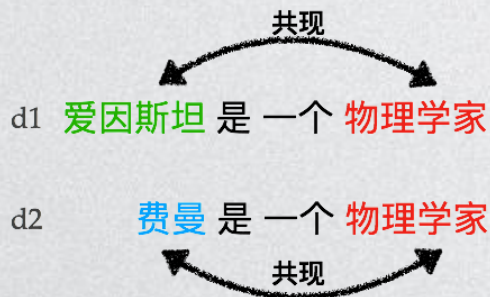


- 输入: 文档集合 (**Bag of words表达**)
- 目标: 发现文档中“潜在的”话题
- 输出
  - 潜在话题 (不同权重的词)
  - 用话题表达的文档



# 文本表示1: TF-IDF、SVD

- 更多的话题模型
  - 采用不同的方式建模**单词在文档中的共现**关系
    - 基于**矩阵分解**的方法，如：LSI等
    - 基于**概率图**的方法，如：PLSI、LDA（最广为使用的话题模型）等



单词-文档共现矩阵

	d1	d2
爱因斯坦	1	0
费曼	0	1
物理学家	1	1