



# 文本表示1: NMF



覃雄派

# 提纲

- NMF定义
- NMF解法
- NMF的结果的理解/与LSI对比
- NMF实践

文本表示1: NMF



# 文本表示1：TF-IDF、SVD

- 《自然》杂志于1999年刊登了D. D. Lee和H. S. Seung非负矩阵分解研究的突出成果
  - 论文提出了一种新的矩阵分解思想，即非负矩阵分解(Non-negative Matrix Factorization, NMF)算法，NMF是在矩阵中所有元素均为非负数约束条件之下的矩阵分解方法

D. D. Lee and H. S. Seung.  
Learning the parts of objects by  
**non-negative matrix factorization.**  
**Nature**, 401(6755):788-791,  
October 1999

## nature

[Explore content](#) ▾ [About the journal](#) ▾ [Publish with us](#) ▾ [Subscribe](#)

[nature](#) > [letters](#) > [article](#)

[Published: 21 October 1999](#)

### Learning the parts of objects by non-negative matrix factorization

[Daniel D. Lee](#) & [H. Sebastian Seung](#) [✉](#)

[Nature](#) **401**, 788–791 (1999) | [Cite this article](#)

**50k** Accesses | **6865** Citations | **49** Altmetric | [Metrics](#)



# 文本表示1: TF-IDF、SVD

- NMF定义
- 非负矩阵分解(nonnegative matrix factorization, NMF)
  - $D \approx UV^T$
  - 输入矩阵 $D$ 所有的数值非负
  - 输出矩阵 $U$ 和 $V$ 的值也非负
  - 不再要求 $U$  (和 $V$ ) 为正交矩阵





# 文本表示1：TF-IDF、SVD

- 该论文的发表，迅速引起了各个领域中的科研人员的重视
  - 首先，科学研究中的很多大规模数据的分析方法需要通过矩阵形式进行有效处理，NMF则为人们处理大规模数据提供了一种新的途径
    - 通过矩阵分解，一方面将描述问题的矩阵的维数进行削减，另一方面也可以对大量的数据进行压缩和概括
  - 其次，NMF分解算法相较于传统的一些算法而言，具有若干优点，包括实现上的简便性、分解形式和分解结果上的可解释性、以及占用存储空间少等
    - 以前，利用矩阵分解来解决实际问题的分析方法很多，比如PCA(主成分分析)、ICA(独立成分分析)、SVD(奇异值分解)、VQ(矢量量化)等
    - 在这些方法中，原始的大矩阵 $V$ 被近似分解为低秩的 $V=WH$ 形式；这些方法的共同特点是，因子 $W$ 和 $H$ 中的元素可为正或负
      - 在数学上，分解结果中存在负值是正确的
      - 但是在实际问题的分析中，负值往往是没有意义的
        - » 比如，图像数据中不可能有负值的像素点；在文档统计中，负值也是无法解释的

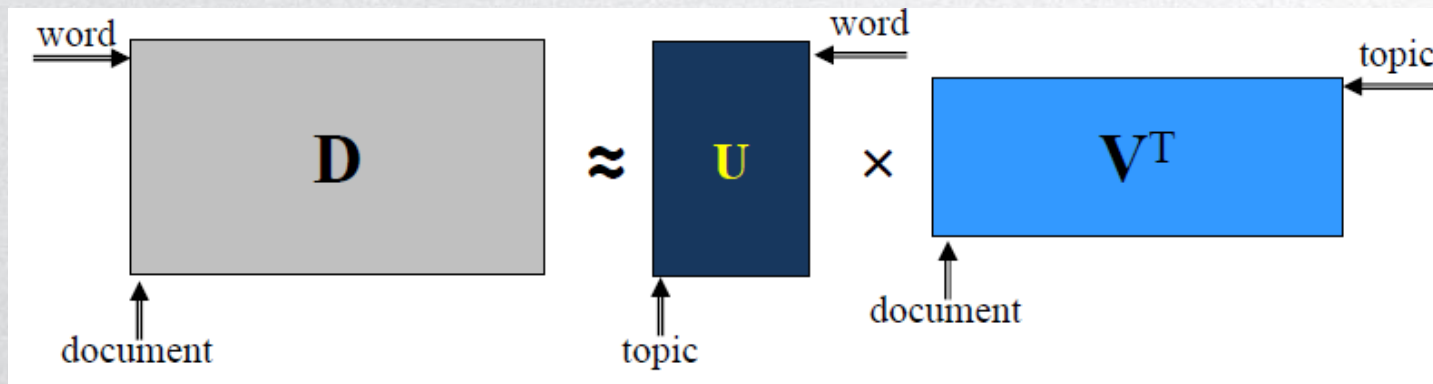
# 文本表示1: TF-IDF、SVD



# 文本表示1: TF-IDF、SVD

- NMF解法
- 非负矩阵分解NMF优化目标

$$\begin{aligned} & - \min_{U,V} |D - U \times V^T|^2 & |D - U \times V^T|^2 &= \sum_{i=1}^M \sum_{j=1}^N (d_{ij} - \mathbf{u}_i \mathbf{v}_j^T)^2 \\ & - \text{s.t. } u_{ij} \geq 0; v_{ij} \geq 0 \end{aligned}$$







# 文本表示1: TF-IDF、SVD

- NMF解法：一般策略
  - 目标函数**非凸**，无全局最优解
  - 在**固定U（或者V）**后，目标函数对V（或者U）是凸函数
  - **交替优化**
    - 1. 随机对U赋值
    - 2. **固定U，最优化V**
    - 3. **固定V，最优化U**
    - 4. 重复2、3，直至收敛





# NMF算法

- 输入:  $D_{M \times N}$ , 隐空间维度  $K$
- 输出:  $U, V$

NMF有不同解法; 这里介绍  
Multiplicative update算法; 具  
体推导看PPT末尾

```

1.  $U \leftarrow$  random nonnegative values
2. repeat
3.   for each  $v_{kj} \in V$ 
4.      $v_{kj} = v_{kj} \frac{[U^T D]_{kj}}{[U^T U V]_{kj}}$ 
5.   end for
6.   for each  $u_{ik} \in U$ 
7.      $u_{ik} = u_{ik} \frac{[D V^T]_{ik}}{[U V V^T]_{ik}}$ 
8.   end for
9. until converge
10. return  $U, V$ 
  
```

$$D_{M \times N}$$

$$U_{M \times K}$$

$$V_{K \times N}$$

$$\frac{K * M \times M * N}{K * M \times M * K \times K * N}$$

$$\frac{M * N \times N * K}{M * K \times K * N \times N * K}$$

# 文本表示1: TF-IDF、SVD



# 文本表示1: TF-IDF、SVD

- NMF解法的推导
- 假设我们把V分解为W和H
  - $L = \min \frac{1}{2} ||V - WH||^2$
  - 计算梯度, 准备利用梯度下降进行法对W和H进行求解
  - $w_{ik} = w_{ik} - \mu_{ik} \frac{\partial L}{\partial w_{ik}}$
  - $h_{kj} = h_{kj} - \mu_{kj} \frac{\partial L}{\partial h_{kj}}$ 
    - 其中,  $\frac{\partial L}{\partial w_{ik}} = -[(V - WH)H^T]_{ik}$
    - $\frac{\partial L}{\partial h_{kj}} = -[W^T(V - WH)]_{kj}$

# 文本表示1: TF-IDF-SVD

- 假设我们把V分解为W和H。

- $L = \min \frac{1}{2} ||V - WH||^2$
- 计算梯度, 准备利用梯度下降进行优化
- $w_{ik} = w_{ik} - \mu_{ik} \frac{\partial L}{\partial w_{ik}}$
- $h_{kj} = h_{kj} - \mu_{kj} \frac{\partial L}{\partial h_{kj}}$

• 其中  $\frac{\partial L}{\partial w_{ik}} = -[(V - WH)H^T]_{ik}$

•  $\frac{\partial L}{\partial h_{kj}} = -[W^T(V - WH)]_{kj}$

相等

- 此处, 仅仅验证一下公式
- 具体证明请参考PPT末尾

0.  $V = \begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix}$ ,  $W = \begin{bmatrix} 1 & 2 \\ 3 & a \end{bmatrix}$ ,  $H = \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix}$ ,  $w_{22}$  为未知量, 计算  $\frac{\partial L}{\partial w_{22}}$

1. 第一种计算方式

$$\frac{1}{2} ||V - WH||^2 = \frac{1}{2} \left\| \begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 3 & a \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix} \right\|^2$$

$$= \frac{1}{2} \left\| \begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix} - \begin{bmatrix} 13 & 16 \\ 9 + 5a & 12 + 6a \end{bmatrix} \right\|^2$$

$$= \frac{1}{2} \text{常数} + \frac{1}{2} (6 - 9 - 5a)^2 + \frac{1}{2} (5 - 12 - 6a)^2$$

对a求导数得到,  $(-3 - 5a)(-5) + (-7 - 6a)(-6) =$

$$15 + 25a + 42 + 36a$$

2. 第二种计算方式

$$-[(V - WH)H^T] = -\left( \begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 3 & a \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix} \right) \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix}$$

$$= -\left( \begin{bmatrix} 8 & 7 \\ 6 & 5 \end{bmatrix} - \begin{bmatrix} 13 & 16 \\ 9 + 5a & 12 + 6a \end{bmatrix} \right) \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix}$$

$$= -\begin{bmatrix} -5 & -9 \\ -3 - 5a & -7 - 6a \end{bmatrix} \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix}$$

$$= -\begin{bmatrix} -15 - 36 & -25 - 54 \\ -9 - 15a - 24 - 24a & -15 - 25a - 42 - 36a \end{bmatrix}$$





# 文本表示1: TF-IDF、SVD

- NMF解法的推导
- 假设我们把V分解为W和H

$$- \frac{\partial L}{\partial w_{ik}} = -[(V - WH)H^T]_{ik}$$

$$- \frac{\partial L}{\partial h_{kj}} = -[W^T(V - WH)]_{kj}$$

- 采用传统的梯度下降法，对于无约束的优化问题，是不能保证结果都是非负的

# 文本表示1: TF-IDF、SVD

- 假设我们把V分解为W和H。

- $L = \min \frac{1}{2} ||V - WH||^2$

- 在这里引入一个技巧, 把减法和加法, 转化成乘法和除法

- $w_{ik} = w_{ik} \frac{[VH^T]_{ik}}{[WHH^T]_{ik}}$

- $h_{kj} = h_{kj} \frac{[W^TV]_{kj}}{[W^TWH]_{kj}}$

注意反方向修改

$$\frac{\partial L}{\partial w_{ik}} = -[(V - WH)H^T]_{ik}$$

$$\frac{\partial L}{\partial h_{kj}} = -[W^T(V - WH)]_{kj}$$

- 这样一来, 如果原来的矩阵是非负的, 那么W、H初始值也非负, 那么结果从始至终都是非负的, 迭代直到满足收敛条件即可

# 文本表示1: TF-IDF、SVD

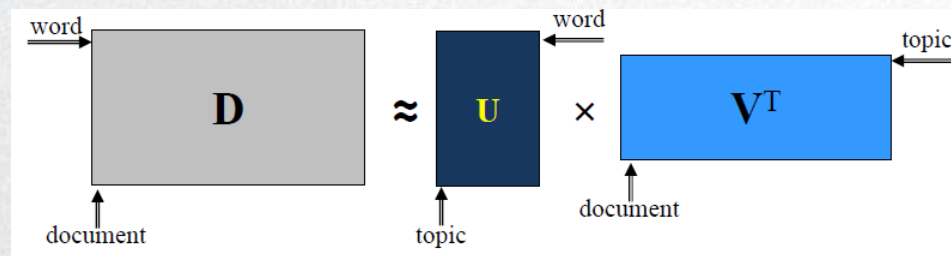




# 文本表示1: TF-IDF、SVD

- NMF小结

- 要求输入/输出矩阵所有的元素都是非负
  - 满足很大一部分文本分析任务
- 优化二次损失函数
  - 无全局最优解
- U和V矩阵可解释
  - D: 词项-文档矩阵
  - U: word和topic的关系
  - V: document和topic的关系





# 文本表示1: TF-IDF、SVD



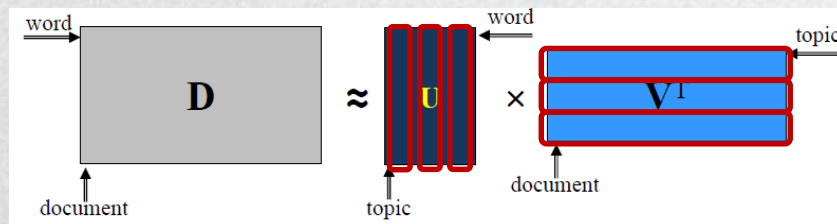
# 文本表示1: TF-IDF、SVD

## • NMF的输出解读

- $U = [u_1, \dots, u_K]$  , 也对应着**K个簇**
- $u_i \in R^M$  ,  $M$ 为集合中不同单词的数目
- $u_i^{(j)}$  : **第j个单词**属于**第i个簇**的“强度”
- $V = [v_1, \dots, v_K]$  , 对应着**K个簇**
- $v_i \in R^N$  ,  $N$ 为集合中文档的数目
- $v_i^{(j)}$  : **第j个文档**属于**第i个簇**的“强度”

- NMF和LSI一样, 可以对单词和文档同时进行“软”聚类
- $K$ 依然很难设定

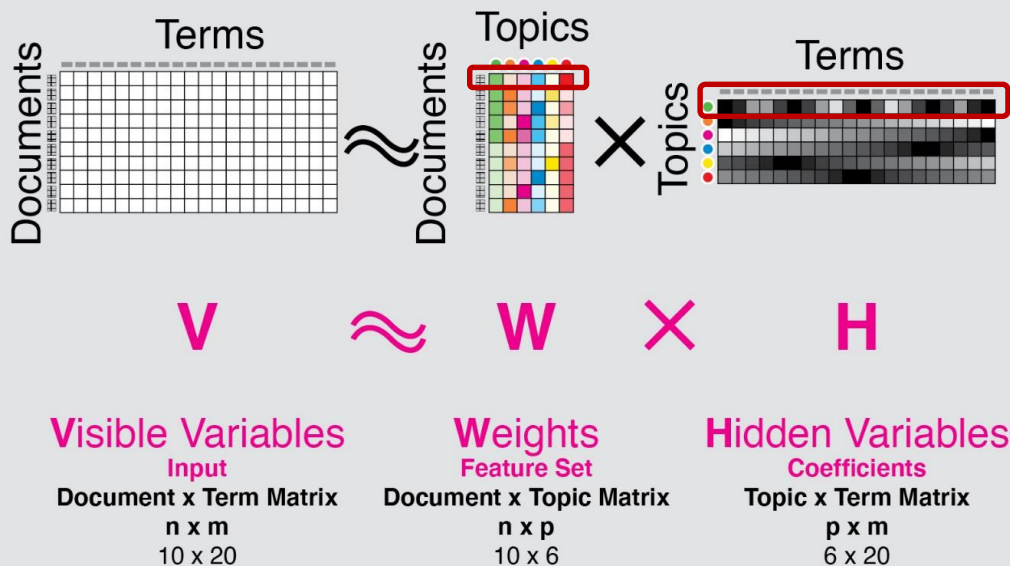
注意这里是  
 $V$ 的转置



# 文本表示1: TF-IDF、SVD

- NMF解读
  - 这里是doc-word矩阵
  - 文档由topic构成
  - Topic由word构成

Non-Negative Matrix Factorization Diagram - Example



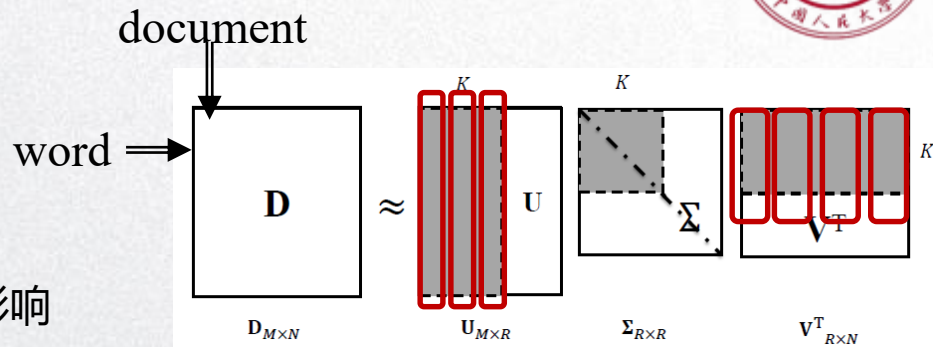


# 文本表示1: TF-IDF、SVD

## • 回顾：理解LSI的输出

- 矩阵  $U_{M \times K} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]$
- $\mathbf{u}_k$ : 第k个话题，由带权的词向量组成
- 由于对应的特征根  $\sigma_i$  为0
- 其余  $N - K$  个特征向量对计算不造成影响

- 矩阵  $V_{K \times N}^T = [v_1 \quad v_2 \quad \dots \quad v_N]$
- $v_n$ : 第n个文档，由k个话题表达



注意这里是v的转置

- LSI的原理：发现词-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





# 文本表示1: TF-IDF、SVD

- NMF实践
  - 装载数据集

## prepare data

```
➤ import numpy as np
  from sklearn.datasets import fetch_20newsgroups
  import matplotlib.pyplot as plt

  %matplotlib inline
  np.set_printoptions(suppress=True)

➤ categories = ['alt.atheism', 'talk.religion.misc', 'comp.graphics', 'sci.space']
  remove = ('headers', 'footers', 'quotes')
  newsgroups_train = fetch_20newsgroups(subset='train', categories=categories, remove=remove)
  newsgroups_test = fetch_20newsgroups(subset='test', categories=categories, remove=remove)
```



# 文本表示1: TF-IDF、SVD

- NMF实践
  - TfidfVectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer_tfidf = TfidfVectorizer(stop_words='english')
vectors_tfidf = vectorizer_tfidf.fit_transform(newsgroups_train.data).todense() # (documents, vocab)
vectors_tfidf.shape
```

(2034, 26576)

```
vocab = np.array(vectorizer_tfidf.get_feature_names())
print("vocab", vocab)
print("vocab shape", vocab.shape)
```

```
vocab ['00' '000' '0000' ... 'zware' 'zwarte' 'zyxel']
vocab shape (26576,)
```

词汇表包含26567个词汇

# 文本表示1: TF-IDF、SVD

- NMF实践

- 取得从大到小的
- id list

```
#test
my_list = [ 1, 37, 5, 57, 9, 8, 6, 5, 3]
my_list = np.asarray(my_list)

id_list = my_list.argsort()[::-5 - 1:-1]
print(id_list)
#取得降序排列的前5个id
```

```
[3 1 4 5 6]
```

- 取得每个topic的
- Top K words

```
import pandas as pd
def get_nmf_topics(vectorizer_tfidf, model, num_topics, n_top_words):
    feat_names = vectorizer_tfidf.get_feature_names() #word list
    word_dict = {};
    for i in range(num_topics):
        words_ids = model.components_[i].argsort()[::-n_top_words - 1:-1]
        words = [feat_names[key] for key in words_ids]
        word_dict['Topic # ' + '{:02d}'.format(i+1)] = words;

    return pd.DataFrame(word_dict);
```



# 文本表示1: TF-IDF、SVD

- NMF实践
  - Run NMF on doc-word matrix
    - 2034doc, 4 topics, 26576 words

```
from sklearn import decomposition  
  
d = 4 # num topics  
clf = decomposition.NMF(n_components=d, random_state=1)
```

```
W1 = clf.fit_transform(vectors_tfidf)  
print(W1.shape)  
H1 = clf.components_  
print(H1.shape)
```

```
(2034, 4)
```

```
(4, 26576)
```

# 文本表示1: TF-IDF、SVD

- NMF实践
  - Show topic
  - Top words

```
num_topics = 4
n_top_words = 8
df2 = get_nmf_topics(vectorizer_tfidf, clf, num_topics, n_top_words)
df2.head(10)
```

	Topic # 01	Topic # 02	Topic # 03	Topic # 04
0	god	graphics	space	ico
1	people	thanks	nasa	bobbe
2	don	files	launch	tek
3	think	image	shuttle	beauchaine
4	jesus	file	moon	bronx
5	just	program	orbit	manhattan
6	say	know	lunar	sank
7	bible	windows	earth	queens

主题分别是：圣经、图形图像、空间NASA、杂项

# 文本表示1: TF-IDF、SVD






# 文本表示1: TF-IDF、SVD


- 梯度计算的证明

矩阵求导相关资料(包括NMF),  
在这个目录下

« 2022newPPT » 0305-回归: 多元线性回归 (解析解与梯度下降算法) »

名称

 misc-矩阵求导.pdf

 misc-矩阵求导.pptx

类型

Adobe Acrobat ...

Microsoft Power...

大小

792 KB

1,256 KB

修改日期

2022/2/10 20:25

2022/2/10 20:01

搜索"0305-回归: 多