# Data Mining and Optimization

## Lecture 4: Keyword Extraction

Liu Yang

Nanjing University

Spring, 2025

# Table of Contents

# 关键词提取简介

- 关键词是代表文章重要内容的一组词。
- 在信息爆炸时代，关键词提取可以大大提高信息获取效率。
- 文本聚类、分类、自动摘要等高级挖掘算法都依赖于关键词提取。

# Table of Contents

# TF-IDF算法

TF-IDF(Term Frequency-Inverse Document Frequency，词频-逆文档频次)算法是一种无监督的关键词提取方法，常用于评估一个文档集中某个词对某份文档的重要程度。一个词对文档越重要，那就越可能是文档的关键词。

- TF算法：统计一个词在一篇文档中出现的频次，其基本思想是，一个词在文档中出现的次数越多，则越重要；
- IDF算法：统计一个词在文档集的多少个文档中出现，其基本思想是，一个词在越少文档中出现，则其对文档的区分能力越强。

# TF-IDF算法

“世界献血日，学校团体、献血服务志愿者等可到血液中心参观检验加工过程，我们会对检验结果进行公示，同时血液的价格也将进行公示。”

- “献血”“血液”“公示”“进行”等词出现的频次均为2，从TF算法角度来看它们对这篇文档的重要性是一样的，但明显“献血”“血液”对这篇文档来说更关键；
- 从IDF算法来看，“进行”在很多文档中都会出现，因此其对文档的区分能力并不强，但“献血”“血液”则在文档集中出现次数不高，具有很强的区分能力。

# TF算法

- 记$n_{ij}$为词$i$在文档$j$中出现的次数，$D$为文档集，$|D|$为文档集中的文档数，$|D_i|$为文档集中包含词$i$的文档数；
- 词$i$在文档$j$中的TF值为：

$$tf_{ij} = \frac{n_{ij}}{\sum_i n_{ij}}. \tag{1}$$

- 长文本中所有词出现的次数都会比短文本更多，但并不表示每个词对长文本都比对短文本更重要；
- 如果只用频次来衡量词的重要性，当比较不同长度文本时，则会得到不合理的结论，因此$tf_{ij}$计算的是词的频率，而非频次。

# IDF算法

- 词$i$的IDF值为：

$$idf_i = log\left(\frac{|D|}{|D_i| + 1}\right). \qquad (2)$$

- 包含词$i$的文档数越少，$idf_i$值越大；
- (2)中分母加1是采用了拉普拉斯平滑，避免有部分新的词在语料库中没有出现而导致分母为零的情况。

# TF-IDF算法

- TF-IDF值是(1)和(2)的乘积：

$$tf - idf_{ij} = \frac{n_{ij}}{\sum_i n_{ij}} log\left(\frac{|D|}{|D_i|+1}\right). \tag{3}$$

- 词$i$的$tf - idf_{ij}$越高，在文档$j$中越重要，越适合作为这篇文档的关键词；
- 一般根据$tf - idf$值的大小排序并选择前$n$个作为关键词。

# TF-IDF算法

TF-IDF算法可以进一步拓展以适应具体的应用场景：

- 添加词性权重：名词作为一种定义现实实体的词，带有更多的关键信息，在计算$tf-idf$值时可赋予更高的权重；
- 添加位置信息：文本起始段落和末尾段落相比其他部分更重要，在计算$tf-idf$值时对出现在这些位置的词可赋予更高的权重。

# Table of Contents

# TextRank算法

与TF-IDF算法不同，TextRank算法不需要一个现成的语料库，仅对单篇文档进行分析就可以提取该文档的关键词。

- TextRank算法的基本思想源于Google的PageRank算法；
- Google创始人Larry Page和Sergey Brin于1997年构建早期搜索系统原型时提出的链接分析算法；
- PageRank是一种网页排名算法，其基本思想为：
  - 链接数量：一个网页被越多的其他网页链接，说明这个网页越重要；
  - 链接质量：一个网页被一个越高权值的网页链接，也能说明这个网页越重要。

# TextRank算法

- 记$In(V_i)$为网页$V_i$的入链集合，$Out(V_j)$为网页$V_j$的出链集合，$|Out(V_j)|$为出链的数量，$S(V_i)$和$S(V_j)$分别表示网页$V_i$和$V_j$的分数(重要性)；
- 每个网页要将它的分数平均地贡献给每个出链，将$V_i$的所有入链贡献给它的分数全部加起来，就是$V_i$自身的分数，即

$$S(V_i) = \sum_{j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|}. \tag{4}$$

# TextRank算法

- (4)式表明，每个网页的分数都与其链接网页的分数有关，那么其链接网页的分数又等于多少呢？
- 为了解决这个问题，算法开始时会将所有网页的分数初始化为1，然后通过多次迭代计算(4)式重新计算所有网页的分数直到算法收敛，收敛时的分数就是网页的最终分数；
- 对于孤立网页(没有出链入链的网页)，上述算法得到的分数为0，则这些网页就不会出现在搜索结果中；
- 为避免这种情况出现，可在(4)式中加入一个阻尼系数$d \in (0,1)$，改进的公式如下：

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|} \tag{5}$$

这样即使一个网页是孤立网页，其得分也会等于$1-d > 0$。

# TextRank算法

- PageRank算法是计算网页的重要性，TextRank算法则是计算一篇文档中词的重要性，为此，将PageRank算法(5)式中的网页换成词即可：

$$S(W_i) = (1-d) + d \sum_{j \in In(W_i)} \frac{S(W_j)}{|Out(W_j)|} \tag{6}$$

其中，$W_i$为词$i$，$In(W_i)$是词$i$的入链集合，$Out(W_j)$是词$j$的出链集合；

- 如何定义一篇文档中词和词之间的链接关系是TextRank算法的关键？

# TextRank算法

- TextRank算法需要一个“窗口”概念，并假设窗口中的词相互间都有链接关系；
- 仍以下面的文本为例：
  “世界献血日，学校团体、献血服务志愿者等可到血液中心参观检验加工过程，我们会对检验结果进行公示，同时血液的价格也将进行公示。”
- 如果将窗口大小设为5，则可得到如下几个窗口：
  - (世界，献血，日，学校，团体)
  - (献血，日，学校，团体，献血)
  - (日，学校，团体，献血，服务)
  - 等

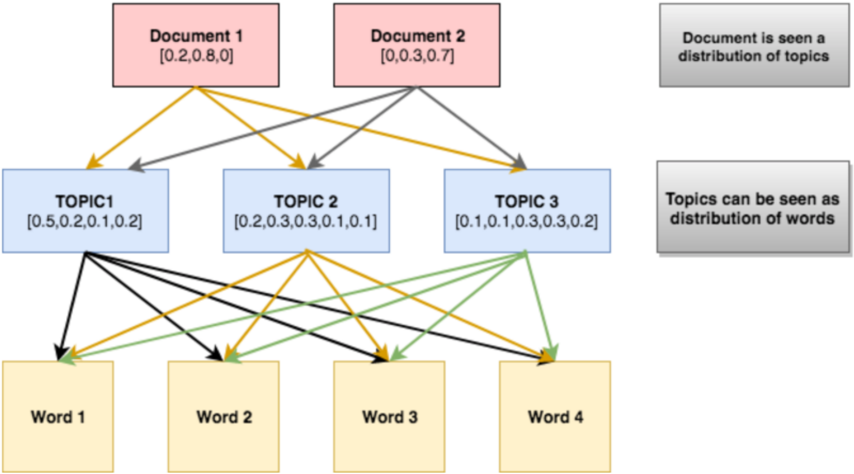
  每个窗口内所有词之间都有链接关系，如“世界”和("献血", "日", "学校", "团体")之间都有链接关系。

# Table of Contents

# 主题模型

- TF-IDF算法和TextRank算法都是基于文档本身的关键词提取，但有些关键词并不一定会显示地出现在文档中；
- 例如一篇讲动物生存环境的科普文，通篇介绍了狮子老虎鳄鱼等各种动物的情况，但是文中并没有出现“动物”一词，此时基于文档本身的关键词提取显然不能提取出“动物”这个隐含的主题信息。

# 主题模型


Document 1
[0.2,0.8,0]
Document 2
[0,0.3,0.7]
Document is seen a distribution of topics
TOPIC1
[0.5,0.2,0.1,0.2]
TOPIC 2
[0.2,0.3,0.3,0.1,0.1]
TOPIC 3
[0.1,0.1,0.3,0.3,0.2]
Topics can be seen as distribution of words
Word 1
Word 2
Word 3
Word 4

# 单词向量空间模型

- 给定一个含有$n$个文本的集合$D=\{d_1,d_2,\cdots,d_n\}$以及文本中出现的所有$m$个单词的集合$W=\{w_1,w_2,\cdots,w_m\}$；
- 将每个单词在每个文本中出现的权重用一个$m\times n$矩阵表示出来，记为

$$X=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}, \tag{7}$$

  其中，$x_{ij}$表示单词$w_i$在文本$d_j$中的权重(tf-idf值)；
- 由于单词的种类很多，而每个文本中出现单词的种类通常较少，所以$X$是一个稀疏矩阵(sparse matrix)。

# 单词向量空间模型

- 对$j = 1, 2, \cdots, n$, $X$矩阵第$j$列的向量表示文本$d_j$，记为

$$x_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{pmatrix}$$

- 整个矩阵$X$表示为

$$X = (x_1, x_2, \cdots, x_n)$$

# 主题向量空间模型

- 一篇文档的语义主要体现在其讨论的主题，两篇文档的语义相近性主要体现在主题相近性；
- 一篇文档一般含有若干个主题，其中某些主题讨论较多，权重较高，另一些主题则较少涉及，权重较低；
- 主题可以由若干个语义相关的单词表示，同义词(如“airplane”与“aircraft”)可以表示同一个话题，而多义词(“apple”)可以表示不同的话题。

# 主题向量空间模型

- 假设所有文本集合$D$共包含$k$个主题，每个主题由一个定义在单词集合$W$上的$m$维向量表示，称为话题向量，即

$$t_l = \begin{pmatrix} t_{1l} \\ t_{2l} \\ \vdots \\ t_{ml} \end{pmatrix}$$

其中$l = 1, 2, \cdots, k$，$t_{il}$表示单词$w_i$在话题$t_l$中的权重，其值越大，该单词在该话题中的重要程度就越高。

# 主题向量空间模型

- 主题向量空间$T$可表示为一个矩阵，称为单词主题矩阵(word-topic matrix)，记作

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1k} \\ t_{21} & t_{22} & \cdots & t_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mk} \end{pmatrix} \tag{8}$$

或

$$T = (t_1, t_2, \cdots, t_k)$$

# 主题向量空间模型

- 现在考虑文本集合$D$中的文本$d_j$，在单词向量空间中由一个向量$x_j$表示，将$x_j$投影到主题向量空间$T$中，得到在主题向量空间的一个向量$y_j$，$y_j$是一个$k$维向量，其表达式为

$$y_j = \begin{pmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{kj} \end{pmatrix}$$

其中$j = 1, 2, \cdots, n$，$y_{lj}$表示主题$t_l$在文本$d_j$中的权重，其值越大，该主题在该文本中的重要程度就越高。

# 主题向量空间模型

- 矩阵$Y$表示主题在文本中出现的权重，称为主题文本矩阵(topic-document matrix)，记作:

$$Y = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{pmatrix}, \tag{9}$$

或

$$Y = (y_1, y_2, \cdots, y_n)$$

# 潜语义分析

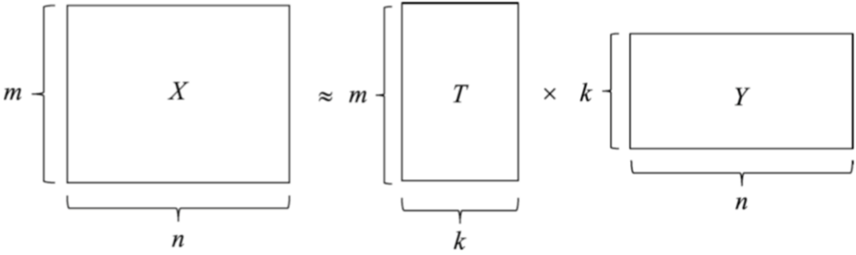
- 在单词向量空间的文本向量$x_j$可以通过它在主题空间中的向量$y_j$近似表示，即由$k$个主题向量以$y_j$为系数的线性组合近似表示

$$x_j \approx y_{1j}t_1 + y_{2j}t_2 + \cdots + y_{kj}t_k$$

  所以，单词文本矩阵$X$可以近似表示为单词主题矩阵$T$与主题文本矩阵$Y$的乘积形式，即$X \approx TY$;
- 实现上述过程的算法又称为潜语义分析(Latent Semantic Analysis, LSA).

# 潜语义分析

直观上，潜语义分析是将文本在单词向量空间的表示通过线性变换转换为在主题向量空间中的表示。



# Table of Contents

# 奇异值分解

- 要进行潜语义分析，需要同时决定两部分的内容，一是主题向量空间$T$，二是文本在主题空间的表示$Y$，使两者的乘积是原始矩阵数据$X$的近似；
- 潜语义分析对单词文本矩阵进行奇异值分解(Singular Value Decomposition，SVD)，将其左矩阵作为主题向量空间，将其对角矩阵与右矩阵的乘积作为文本在主题向量空间的表示。

# 奇异值分解

## Singular Value Decomposition

任意一个非零的$m \times n$实矩阵$A$可表示为如下形式:

$$A = U\Sigma V', \tag{10}$$

其中，$U$是$m$阶正交矩阵(orthogonal matrix)，$V$是$n$阶正交矩阵，$\Sigma$是由降序排列的非负对角线元素组成的$m \times n$矩形对角矩阵(rectangular diagonal matrix)满足

$$\begin{aligned} UU' &= I \\ VV' &= I \\ \Sigma &= diag\{\sigma_1, \sigma_2, \cdots, \sigma_p\} \\ \sigma_1 &\geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0 \\ p &= \min\{m, n\}. \end{aligned} \tag{11}$$

# 奇异值分解

- $U\Sigma V'$：矩阵$A$的奇异值分解(singular value decomposition，SVD)；
- $\sigma_i$：矩阵$A$的奇异值(singular value)；
- $U$的列向量：左奇异向量(left singular vector)；
- $V$的列向量：右奇异向量(right singular vector)；
- 注意奇异值分解不要求矩阵$A$是方阵，事实上矩阵的奇异值分解可以看作是方阵对角化的推广；
- 矩阵的奇异值分解一定存在但不一定唯一。

# 奇异值分解的例子

- 给定一个$5 \times 4$的矩阵$A$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

# 奇异值分解的例子

- $A$的奇异值分解为$U\Sigma V'$，其中，

$$U = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & 0 & \sqrt{0.8} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \sqrt{0.8} & 0 & -\sqrt{0.2} \end{pmatrix}$$

且$UU' = I_5$.

# 奇异值分解的例子

- $A$的奇异值分解为$U\Sigma V'$，其中，

$$V' = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

且$VV' = I_4$.

# 奇异值分解的例子

- $A$的奇异值分解为$U\Sigma V'$，其中，

$$\Sigma = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

.

# 奇异值分解的例子

- $A$的奇异值分解不是唯一的。在此例中，如果选择

$$U = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & \sqrt{0.4} & -\sqrt{0.4} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{0.5} & \sqrt{0.5} \\ 0 & 0 & \sqrt{0.8} & -\sqrt{0.1} & \sqrt{0.1} \end{pmatrix}$$

而$\Sigma$和$V$不变，那么$U\Sigma V'$也是$A$的一个奇异值分解。

# 奇异值分解的性质

- 矩阵$A$的奇异值分解$U\Sigma V'$中，$\sigma_1, \sigma_2, \cdots, \sigma_p$是唯一的，而矩阵$U$和$V$不是唯一的；
- 矩阵$A$的秩和$\Sigma$的秩相等，等于正奇异值$\sigma_i$的个数$r$（包含重复的奇异值）。

# 截断奇异值分解

- (10)式中，$A = U\Sigma V'$叫做完全奇异值分解(full singular value decomposition)；
- 实际使用的奇异值分解为截断奇异值分解(truncated singular value decomposition)。

# 截断奇异值分解

**Truncated Singular Value Decomposition**

记$A$为非零的$m \times n$实矩阵，其秩$rank(A) = r$，且$0 < k < r$，则称$U_k \Sigma_k V_k'$为矩阵$A$的截断奇异值分解

$$A \approx U_k \Sigma_k V_k', \tag{12}$$

其中，$U_k$是$m \times k$阶矩阵，$V_k$是$n \times k$阶矩阵，$\Sigma_k$是$k$阶对角矩阵，矩阵$U_k$由完全奇异值分解中$U$的前$k$列组成，矩阵$V_k$由$V$的前$k$列组成，矩阵$\Sigma_k$由$\Sigma$的前$k$个对角线元素得到，因此，对角矩阵$\Sigma_k$比原始矩阵$A$的秩低。

截断奇异值分解得到的矩阵$U_k \Sigma_k V_k'$的秩为$k$，通常远小于原始矩阵的秩$r$，从而实现了由低秩矩阵对原始矩阵的压缩，也是在秩不超过$k$的$m \times n$矩阵中对原始矩阵$A$的一个最优近似。

# 截断奇异值分解的例子

- 给定一个$5 \times 4$的矩阵$A$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

# 截断奇异值分解的例子

- $A$的秩为3，若取$k=2$，则

$$U_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix}, V_2' = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

因此，$A$的截断奇异值分解为

$$U_2 \Sigma_2 V_2' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

# Table of Contents

# 弗罗贝尼乌斯范数

- 矩阵$A$的截断奇异值分解$U_k\Sigma_k V_k'$是对$A$的一种近似方法，这个近似是在弗罗贝尼乌斯范数(Frobenius norm)意义下的近似。

**Frobenius norm**

对任意$m \times n$矩阵$A = [a_{ij}]_{m\times n}$，其Frobenius norm为

$$\|A\|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \right)^{\frac{1}{2}}. \tag{13}$$

**矩阵最优近似**

记$A$为非零的$m \times n$实矩阵，其秩$rank(A) = r$，记$\mathbb{M}_k$为所有秩不超过$k$的$m \times n$实矩阵的集合，且$0 < k < r$，则

$$\|A - A'\|_F = \min_{S \in \mathbb{M}_k} \|A - S\|_F, \tag{14}$$

其中，$A' = U_k \Sigma_k V_k'$为(12)式中矩阵$A$的截断奇异值分解。

# Table of Contents

# 单词文本矩阵

- 给定文本集合$D=\{d_1,d_2,\cdots,d_n\}$以及单词集合$W=\{w_1,w_2,\cdots,w_m\}$。首先将数据表示为单词文本矩阵

$$X=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix};$$

- 确定主题个数$k$并对$X$进行截断奇异值分解

$$X\approx U_k\Sigma_k V_k'=(u_1,u_2,\cdots,u_k)\begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix}\begin{pmatrix} v_1' \\ v_2' \\ \vdots \\ v_k' \end{pmatrix}.$$

# 主题向量

- 在单词文本矩阵$X$的截断奇异值分解式中，矩阵$U_k$的列向量$u_1, u_2, \cdots, u_k$表示$k$个主题，称为主题向量；
- 由这$k$个主题向量张成一个子空间，$U_k = (u_1, u_2, \cdots, u_k)$称为主题向量空间。

# 主题文本矩阵

- 有了主题向量空间，接着考虑文本在主题向量空间的表示

$$
\begin{aligned}
X &= (x_1, x_2, \cdots, x_n) \approx U_k \Sigma_k V_k' \\
&= (u_1, u_2, \cdots, u_k) \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & \cdots & v_{n1} \\ v_{12} & v_{22} & \cdots & v_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1k} & v_{2k} & \cdots & v_{nk} \end{pmatrix} \\
&= \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1k} \\ u_{21} & u_{22} & \cdots & u_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mk} \end{pmatrix} \begin{pmatrix} \sigma_1 v_{11} & \sigma_1 v_{21} & \cdots & \sigma_1 v_{n1} \\ \sigma_2 v_{12} & \sigma_2 v_{22} & \cdots & \sigma_2 v_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_k v_{1k} & \sigma_k v_{2k} & \cdots & \sigma_k v_{nk} \end{pmatrix}
\end{aligned}
$$

# 主题文本矩阵

- 矩阵$X$的第$j$列$x_j$满足

$$x_j \approx U_k(\Sigma_k V_k')_j = (u_1, u_2, \cdots, u_k)\begin{pmatrix} \sigma_1 v_{j1} \\ \sigma_2 v_{j2} \\ \vdots \\ \sigma_k v_{jk} \end{pmatrix} = \sum_{l=1}^{k} \sigma_l v_{jl} u_l, j = 1, 2, \cdots, n,$$

其中，$(\Sigma_k V_k')_j$是矩阵$\Sigma_k V_k'$的第$j$列，即文本$d_j$可由$k$个主题向量$u_l$的线性组合近似表示。

# 主题文本矩阵

- 矩阵$\Sigma_k V_k'$的每一个列向量

$$\begin{pmatrix} \sigma_1 v_{11} \\ \sigma_2 v_{12} \\ \vdots \\ \sigma_k v_{1k} \end{pmatrix}, \begin{pmatrix} \sigma_1 v_{21} \\ \sigma_2 v_{22} \\ \vdots \\ \sigma_k v_{2k} \end{pmatrix}, \cdots, \begin{pmatrix} \sigma_1 v_{n1} \\ \sigma_2 v_{n2} \\ \vdots \\ \sigma_k v_{nk} \end{pmatrix}$$

  分别表示一个文本在主题向量空间的表示；
- 综上，可以通过对单词文本矩阵的奇异值分解$X \approx U_k \Sigma_k V_k'$，得到主题空间$U_k$，以及文本在主题空间的表示$\Sigma_k V_k'$。