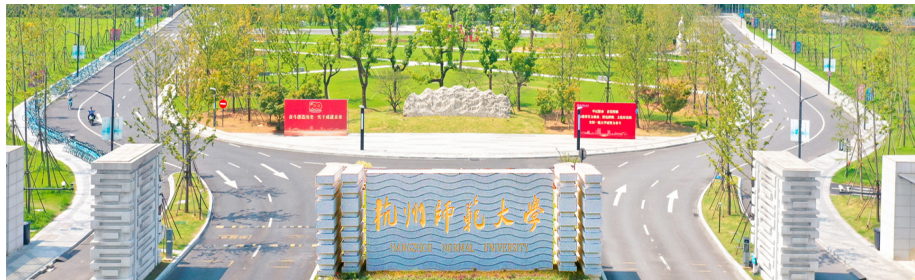


# 第六讲 - 主题模型

张建章

阿里巴巴商学院  
杭州师范大学

2024-03-01



- 1 文本主题建模介绍
- 2 潜在语义索引
- 3 概率潜在语义分析
- 4 潜在狄利克雷分配 (LDA)
- 5 主题模型评价指标
- 6 新闻文本主题建模实例
- 7 课后实践

# 1. 文本主题建模介绍

文本主题建模 (Topic Modeling): 是一种无监督文本挖掘技术, 通过挖掘文档中词语的共现模式, 将其映射到一组潜在的主题中, 以自动发现和抽取文档集合的主题结构, 更好地了解文档集合信息。

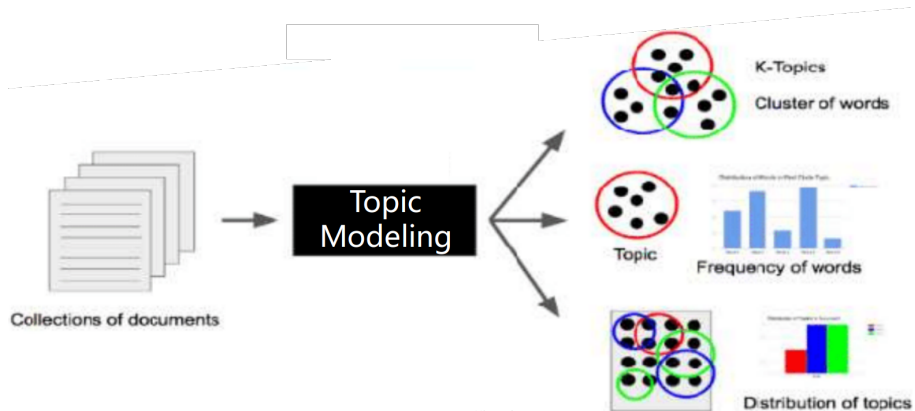


图 1: 主题建模示意图

常用方法有潜在语义索引 (LSI)、概率潜在语义分析 (pLSA) 和潜在狄利克雷分配 (LDA)。



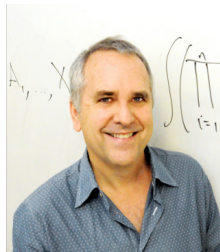
Thomas Hofmann



David M. Blei



Andrew Ng



Michael I. Jordan

[1] Hofmann, Thomas. “Probabilistic latent semantic indexing.” Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999.

[2] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. “Latent dirichlet allocation.” Journal of machine Learning research 3, no. Jan (2003): 993-1022.

## 主题建模与文本聚类的区别

都是无监督文本挖掘方法，从大量文档中挖掘隐藏的结构和主题，主要区别如下：

- **目标：**文本聚类关注将文档分组，而主题模型关注发现潜在的主题结构。
- **输出：**文本聚类的输出是离散的聚类标签，而主题模型的输出是连续的文档-主题分布和主题-单词分布。
- **方法：**文本聚类通常基于距离或相似度度量，而主题模型通常基于概率模型。

在某些情况下可以互相补充，例如，可以使用主题模型的输出（文档-主题分布）作为文本聚类的输入，以发现更具语义信息的簇。

## 简介

潜在语义索引/分析 (latent semantic indexing/analysis, LSI/LSA): 是一种基于线性代数技术的主题建模方法, 使用奇异值分解 (SVD) 对词语-文档矩阵进行降维, 捕捉文档和词项之间的潜在语义结构。

词汇-文档矩阵 (TF-IDF)

	Doc-1	Doc-2	Doc-3	Doc-4
Term-1				
Term-2				
Term-3				
Term-4				

 $M \times N$ 

词汇-主题矩阵

	Topic-1	Topic-2
Term-1		
Term-2		
Term-3		
Term-4		

 $M \times K$ 

主题重要性对角阵

	Topic-1	Topic-2
Topic-1		
Topic-2		

 $K \times K$ 

主题-文档矩阵

	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1				
Topic-2				

 $K \times N$ 

图 2: 潜在语义索引原理示意图

$M$  为文档集的词表规模,  $N$  为文档数量,  $K$  表示主题数 (超参数, 提前指定)。基于矩阵分解而非概率统计技术, 故矩阵元素为实数, 无概率含义, 第一个矩阵可视为词汇在潜在主题空间上的向量表示, 第三个矩阵可视为文档在潜在主题空间上的向量表示, 主题不可解释。

## 原理-奇异值分解

线性代数中的一种矩阵分解技术，将一个复杂的矩阵分解为几个简单的矩阵相乘。

对于一个给定的  $m \times n$  矩阵  $A$ ，SVD 可以将其分解为三个矩阵的乘积：一个  $m \times m$  的正交矩阵  $U$ 、一个  $m \times n$  的对角矩阵（只有对角线上有非零元素） $\Sigma$  和一个  $n \times n$  的正交矩阵  $V$  的转置矩阵  $V^T$ ，即：

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

矩阵  $U$  和  $V$  的列向量是分别是  $A$  的左奇异向量和右奇异向量， $\Sigma$  对角线上的元素是  $A$  的奇异值，它们都是非负实数，并按降序排列。奇异值反映了数据矩阵  $A$  中各个方向上的能量或信息量。

LSI 中通过保留矩阵  $\Sigma$  中前  $k$  个较大的奇异值（及对应的  $U$  和  $V$  的列向量），可以得到原矩阵  $A$  的近似表示，从而实现数据的降维和降噪：

$$A \approx U_{m \times k} \Sigma_{k \times k} V_{k \times n}^T$$

## 简介

概率潜在语义分析 (probabilistic latent semantic indexing/analysis, pLSI/pLSA): 基于概率生成模型的主题建模方法, 使用极大似然估计学习文档、词语和主题之间的联合概率分布, 核心思想是将每个文档视为主题的混合, 每个主题又是词语的概率分布。最终输出两个参数矩阵:

- ① 文档-主题概率分布矩阵  $P(c|d)$ : 行表示文档, 列表示主题, 每个元素表示主题  $c$  在文档  $d$  中的概率。
- ② 主题-词汇概率分布矩阵  $P(w|c)$ : 行表示主题, 列表示词语, 每个元素表示词语  $w$  在主题  $c$  中的概率。





### 3. 概率潜在语义分析

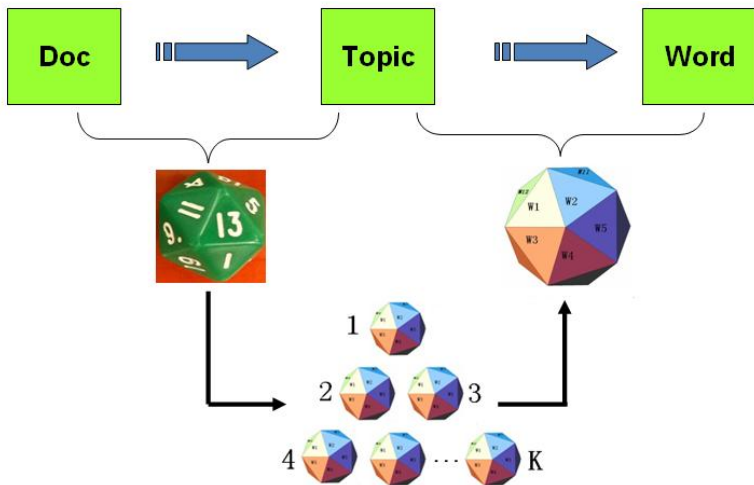


图 3: pLSA 文档生成过程示意图

- 1: 上帝有两种类型的骰子, 一类是doc-topic 骰子, 每个doc-topic 骰子有 $K$  个面, 每个面是一个topic 的编号; 一类是topic-word 骰子, 每个topic-word 骰子有 $V$  个面, 每个面对应一个词;



**doc-topic**



**topic-word**

- 2: 上帝一共有 $K$  个topic-word 骰子, 每个骰子有一个编号, 编号从1 到 $K$ ;
- 3: 生成每篇文档之前, 上帝都先为这篇文章制造一个特定的doc-topic 骰子, 然后重复如下过程生成文档中的词
  - 投掷这个doc-topic 骰子, 得到一个topic 编号 $z$
  - 选择 $K$  个topic-word 骰子中编号为 $z$ 的那个, 投掷这个骰子, 于是得到一个词

图 4: pLSA 生成过程通俗解释

## 原理 - 概率生成模型

**前提：** 将文档集合视为一组以 (词语, 文档) 形式表示的观测，即  $(w, d)$ 。

**假设：** pLSA 将每个共现的概率建模为条件独立的多项式分布的混合：

$$P(w, d) = \sum_c P(c)P(d|c)P(w|c) \quad (1)$$

由条件独立，可得：

$$\sum_c P(c)P(d|c)P(w|c) = \sum_c P(c)P(wd|c)$$

由全概率公式，可得：

$$\sum_c P(c)P(wd|c) = P(w, d)$$

$c$  表示主题，主题数为超参数，需提前指定。

### 3. 概率潜在语义分析

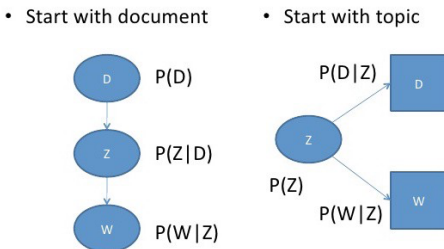
式 (1) 为 pLSA 的对称对称表达式, 即, 词语和文档均由主题以多项式分布生成, 对应条件概率  $P(w|c)$  和  $P(d|c)$ 。

由条件概率公式, 可得 pLSA 的非对称表示形式, 如下:

$$P(w, d) = P(d) \sum_c P(c|d) P(w|c) \quad (2)$$

即, 对于给定的文档  $d$ , 以概率  $P(c|d)$  选择主题  $c$ , 并在主题  $c$  下, 以概率  $P(w|c)$  生成词语  $w$ 。

对称与非对称的 pLSA 表示形式的区别, 如下图 ( $Z$  表示主题) 所示:



**求解：**共  $cd + wc$  个参数，采用期望最大化 (expectation maximization, EM) 算法寻找极大似然估计。

文本挖掘中通常使用 pLSA 的非对称表示形式，其盘式表示法 (plate notation) 如下：

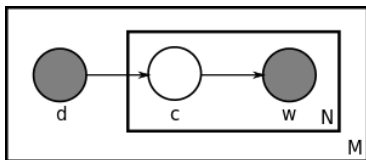


图 5: pLSA 非对称形式的盘式表示法

上图中， $M$  表示文档数， $N$  表示词表规模， $d$  表示单个文档， $c$  表示单个主题， $w$  表示单个词语。

pLSA 基于统计推断，其结果向量中的元素表示概率值，而 LSA 结果向量中的元素值为实数，无概率含义。

## 介绍

**潜在狄利克雷分配 (Latent Dirichlet allocation, LDA):** 目前最流行的主题建模方法之一，是 pLSA 的贝叶斯扩展，使用狄利克雷先验分布对文档-主题和主题-词项的多项式分布进行建模，通过贝叶斯推断来学习这些多项式分布的后验概率。

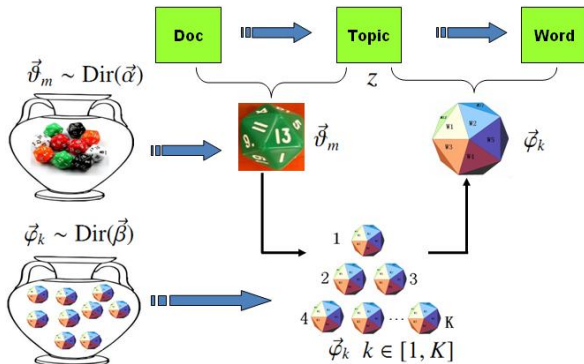
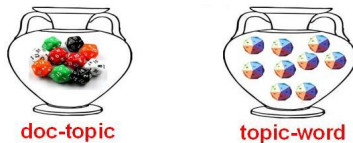


图 6: LDA 文档生成过程示意图

## 4. 潜在狄利克雷分配 (LDA)

- 1: 上帝有两大坛子的骰子, 第一个坛子装的是doc-topic 骰子,第二个坛子装的是topic-word 骰子;



- 2: 上帝随机的从第二个坛子中独立的抽取了 $K$  个topic-word 骰子, 编号为1到 $K$ ;
- 3: 每次生成一篇新的文档前, 上帝先从第一个坛子中随机抽取一个doc-topic 骰子, 然后重复如下过程生成文档中的词
  - 投掷这个doc-topic 骰子,得到一个topic 编号 $z$
  - 选择 $K$  个topic-word 骰子中编号为 $z$ 的那个, 投掷这个骰子, 于是得到一个词

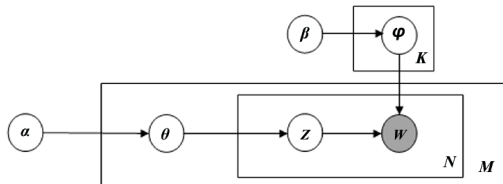
图 7: LDA 文档生成过程通俗解释

## 原理 - 概率生成模型

LDA 对 pLSA 的关键扩展在于使用多项式的共轭先验分布——狄利克雷分布对多项式分布的参数进行贝叶斯参数估计，pLSA 将多项式分布的参数作为常量，而 LDA 将多项式分布的参数作为服从狄利克雷分布的变量，即，把式 (2) 扩展如下：

$$P(w, d) = P(d) \sum_c P(c|d; \theta) P(w|c; \phi) \quad (3)$$

其中  $\theta$  和  $\phi$  分别表示文档-主题多项式分布和主题-词语多项式分布的参数。盘式表示法如下图， $\alpha$  和  $\beta$  分别为文档-主题多项式分布和主题-词语多项式分布的狄利克雷先验参数。





#### 4. 潜在狄利克雷分配 (LDA)

LDA 的前提和假设同 pLSA，参数估计通常采用 Gibbs 采样或变分贝叶斯，LDA 输出结果中的最常用的两个参数矩阵也是文档-主题概率分布矩阵，主题-词语概率分布矩阵。

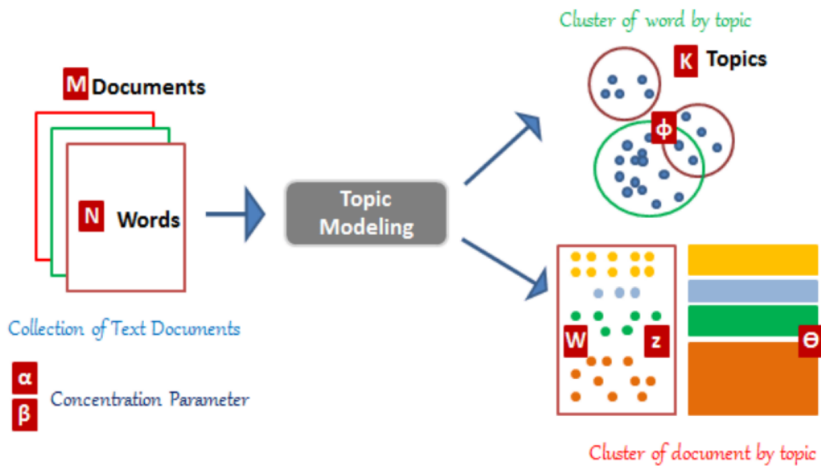


图 8: LDA 文本主题挖掘示意图

## 困惑度 I

困惑度 (Perplexity) 是衡量主题模型性能的常用指标, 反映了模型对新数据的预测能力, 值越低表示模型对新数据的预测能力越强, 可以理解为测试数据的对数似然 (log-likelihood) 的负指数化, 计算原理如下:

$$\text{Perplexity}(D) = \exp \left( -\frac{\sum_{d=1}^{|D|} \log p(w_d)}{N} \right)$$

其中:

- $D$  是测试文档集;
- $|D|$  是测试文档的数量;
- $N$  是所有测试文档中的词语总数;
- $w_d$  是第  $d$  个文档;
- $p(w_d)$  是文档  $d$  的词语的生成概率。

## 困惑度 II

对于 LDA 模型，困惑度可以通过以下方式计算：

1. 计算每个文档的似然：

- 每个文档  $d$  的似然  $p(w_d)$  是文档中所有词语似然的乘积。对于文档  $d$  中的每个词  $w$ ，其似然可以通过 LDA 模型的参数 (文档-主题分布和主题-词分布) 来计算：

对于文档  $d$ ，其生成概率  $p(w_d)$  可以表示为文档中所有词语的联合概率：

$$p(w_d) = \prod_{n=1}^{N_d} p(w_{d,n})$$

每个词语  $w_{d,n}$  的概率  $p(w_{d,n})$  是该词在所有主题中的概率加权平均 (全概率公式)：

## 困惑度 III

$$p(w_{d,n}) = \sum_{k=1}^K p(w_{d,n}|z_{d,n} = k)p(z_{d,n} = k|d)$$

-  $p(w_{d,n}|z_{d,n} = k)$  是词  $w_{d,n}$  在主题  $k$  中的概率，可从主题-词分布矩阵  $\beta$  中获取；

-  $p(z_{d,n} = k|d)$  是主题  $k$  在文档  $d$  中的概率，可从文档-主题分布矩阵中获取；

2. 计算总体似然：

- 总体似然是所有文档似然的和；

3. 平均对数似然：

- 平均对数似然是总体对数似然除以文档中词语的总数  $N$ ；

4. 计算困惑度：困惑度是平均对数似然的负指数化。

### 根据困惑度值选择恰当的主题数 I

#### 困惑度随主题数变化的趋势:

① 通常情况下，困惑度会随着主题数的增加而降低。因为更多的主题数提供了更多的自由度来解释数据，导致模型对训练数据的拟合变得更好；

② 当增加主题数时，困惑度的下降幅度会逐渐减小，找到困惑度下降减缓的拐点可以帮助选择一个合适的主题数；

③ 选择一个困惑度较低但主题数不至于太大的模型，可以平衡模型的复杂度和在新数据上的表现，在选择主题数时，不仅要考虑困惑度，还要考虑主题的解释性，确保选择的主题数能够生成有意义且易于解释的主题。

## 主题一致性 I

主题一致性：评价主题模型生成的主题质量的常用指标，通过计算主题词在文档中的共现频率，评估主题词之间的语义一致性，一种常用的主题一致性指标  $c_v$ <sup>12</sup> 的计算过程如下：

## 1. 提取主题词

从每个主题中提取权重最大的前  $N$  个词。

## 2. 构建词对共现矩阵

使用滑动窗口的方法在原始文档中统计词对的共现频率，共现矩阵记录了词对在一定窗口大小内共同出现的频率。

## 3. 计算点互信息 (PMI)

点互信息 (PMI) 用于衡量词对的共现关联性。对于每对词  $w_i$  和  $w_j$ ，PMI 的计算公式为：

$$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

## 主题一致性 II

- $P(w_i, w_j)$  是词  $w_i$  和词  $w_j$  在窗口内共同出现的概率;
- $P(w_i)$  和  $P(w_j)$  分别是词  $w_i$  和  $w_j$  的出现概率。

## 4. 计算每个主题的一致性得分

对于每个主题，将其所有词对的 PMI 值进行聚合得到主题的一致性得分。具体而言，可以计算每个词对的 PMI 值，然后取平均值或其他统计值。

5. 计算整个模型的一致性得分  $c_v$ 

最终，模型的  $c_v$  值是所有主题一致性得分的平均值。

---

<sup>1</sup>Röder, Michael, Andreas Both, and Alexander Hinneburg. "Exploring the space of topic coherence measures." In Proceedings of the eighth ACM international conference on Web search and data mining, pp. 399-408. 2015.

<sup>2</sup><https://radimrehurek.com/gensim/models/coherencemodel.html>

## 主题差异性

主题差异性衡量主题之间的不同程度，一个简单的度量方式是计算每个主题的前  $n$  个词的唯一词比例。

$$\text{Topic Diversity} = \frac{\text{Number of Unique Words in Top-}n \text{ Terms}}{K \times n}$$

其中， $K$  是主题的数量。



## 人工评价

尽管自动化评价指标很重要，但人类评价仍然是主题模型评价的关键部分。通过人工检查主题词的质量和对实际问题的解释能力，可以获得模型的真实表现。

**总结:** 在实际应用中，通常需要结合多种指标来全面评估 LDA 模型的效果。困惑度和对数似然适用于衡量模型的整体表现，而主题一致性和主题差异性则帮助评估主题的质量和区分度。结合人类评价，可以更准确地选择和调整主题模型。

## 1 - 加载所需软件包和数据集

```
import numpy as np
import re
import gensim
from gensim import corpora
from sklearn.datasets import fetch_20newsgroups
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
import nltk
from nltk.corpus import stopwords

# 加载数据集
newsgroups = fetch_20newsgroups(subset='all', remove=('headers',
↪ 'footers', 'quotes'))
```

## 2 - 预处理文本数据

```
# 下载英文停用词
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
def preprocess(text):
    text = re.sub(r'\W', ' ', text) # 移除非单词字符
    text = re.sub(r'\s+', ' ', text) # 移除多余空格
    text = text.lower().strip()
    words = text.split()
    words = [word for word in words if word not in stop_words and
    ↪ len(word) > 2] # 移除停用词
    return words

documents = [preprocess(document) for document in
    ↪ newsgroups.data]

# 创建字典和语料库
dictionary = corpora.Dictionary(documents)
corpus = [dictionary.doc2bow(document) for document in documents]
```

## 3 - 训练主题模型

```
# 指定主题数量
num_topics = 20

# 训练 pLSA 模型
lsi = gensim.models.LsiModel(corpus, id2word=dictionary,
    ↪ num_topics=num_topics)

# 显示主题 - pLSA
topics = lsi.print_topics(num_topics=num_topics)
for topic in topics:
    print(topic)
```

## 4 - LDA 结果可视化

```
# 训练 LDA 模型
lda = gensim.models.LdaModel(corpus, id2word=dictionary,
    ↪ num_topics=num_topics, random_state=2023)

# 显示主题-LDA
topics = lda.print_topics(num_topics=num_topics)
for topic in topics:
    print(topic)

# 准备 LDavis 数据
lda_vis_data = gensimvis.prepare(lda, corpus, dictionary)

# 可视化
pyLDavis.display(lda_vis_data)
```

## 4 - LDA 结果可视化

从下图右侧的主题-词汇分布可知，该主题为“太空探索”相关。

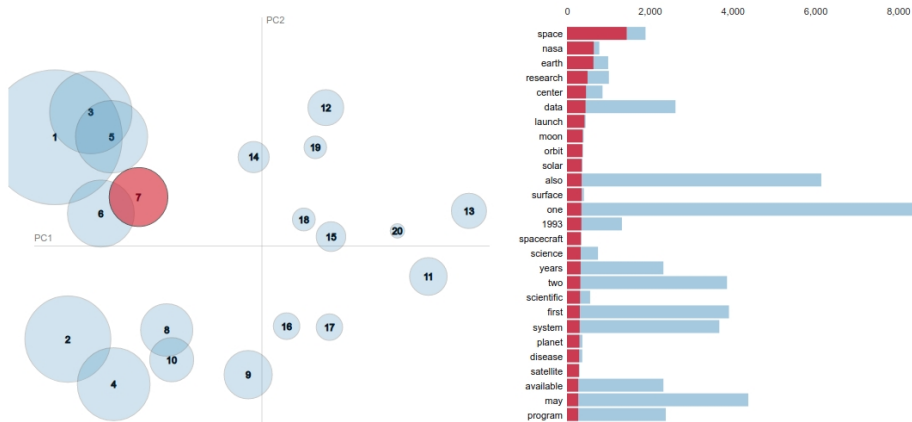


图 9: LDA 主题建模结果可视化示例

1. 使用本课程提供的中文新闻语料库，使用 `gensim` 中的文本主题建模方法进行主题挖掘，具体要求如下：

- ① 自行查找资料学习主题挖掘结果定量度量指标；
- ② 绘图对比不同主题建模方法结果的定量指标，横轴为主题数  $K$ ，纵轴为所选取的定量指标；
- ③ 学习使用 `BERTopic` ([主页](#))包进行主题建模；

THE END