机器学习概论 实验报告

实验题目：Lab5 隐狄里克雷分配模型(LDA)

学生姓名：王章瀚

学生学号：PB18111697

完成日期：2021 年 2 月 19 日

计算机实验教学中心制

2019年09月

目录

1 实验介绍

本实验给定了包含 1000 篇随机抽取的文档的数据集, 这些新闻来自20个不同的主题. 实验要求利用 LDA 模型和吉布斯采样算法, 给出这20个主题相关的 top 10 关键词, 并按照概率大小排序.

2 理论基础

2.1 LDA 模型

隐狄里克雷分配模型(Latent Dirichlet Allocation, LDA) 是话题模型的典型代表.

2.1.1 LDA 相关的基本概念及变量定义

- LDA 的基本单元有:
  - 词 (word): 待处理数据中的基本离散单元. 图片中的一块像素块也能视为一个"词".
  - 文档(document): 待处理的数据对象, 由词构成. 词在文档中以词袋的形式体现, 不计顺序.
  - 话题(topic): 表示一个概念, 具体表现为一系列相关联的词, 以及它们在该概念下出现的频率.
- 假定数据集中共含有 K 个话题和 D 篇文档, 词来自含 V 个词的字典.
- 每篇**文档**用长度为 N_d 的单词表示, 即文档集合为 $\mathcal{D} = \{w_1, w_2, \cdots, w_D\}$, 其中 $w_d = (w_{d1}, w_{d2}, \cdots, w_{dN_d})$ 为第 d 篇文档的单词序列.
- 每个**话题**用长度为 V 的概率词向量 $\boldsymbol{\beta}$ 表示, 且 $\boldsymbol{\beta}_k \in [0,1]^V$. 话题集合为 $Z = \{z_1, \cdots, z_K\}$. $\beta_{kv} = P(w_v|z_k)$ 即表示第 k 个话题中单词 w_v 的概率.
- **文档中话题的分布**记为 $\boldsymbol{\theta}_d \in [0,1]^K, \theta_{dk} = P(z_k|\boldsymbol{w}_d)$.

2.1.2 文档和主题的生成过程

文档和主题的生成过程是按照如下步骤来进行的:

- 生成文档 d 的过程:
  1. 从以 α 为参数的狄利克雷分布中随机采样一个话题分布 θ_d;
  2. 按如下步骤生成文档中的第 N_d 个词:
     I. 根据 θ_d 进行话题指派, 得到文档 d 中第 v 词的话题 z_{dv}
     II. 根据指派的话题 z_{dv} 所对应的词分布 $\boldsymbol{\beta}_k$ 随机采样生成词 w_{dv}
- 生成主题 k 的过程: 从以 η 为参数的狄利克雷分布中随机采样一个话题分布 $\boldsymbol{\beta}_k$

按照以上的步骤我们就可以生成文档和主题. 这相应的示意图如下所示:

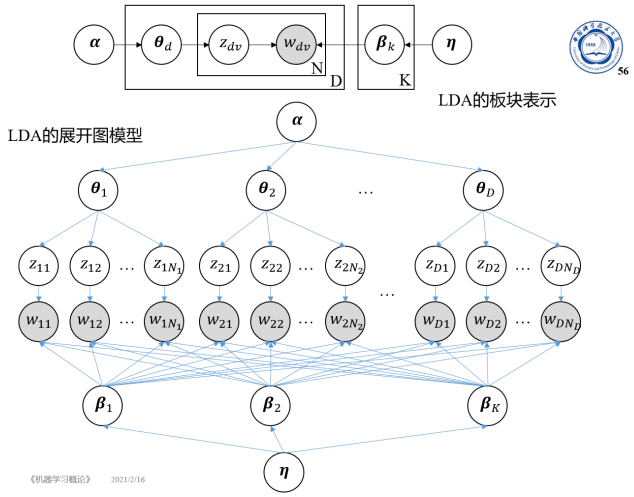

图 1: LDA 图解(图源自连德富老师的 PPT)

2.1.3 LDA 对应概率分布及求解方式

有了上述图解, 很容易可以写出来下式

$$p(W,z,\beta,\Theta|\alpha,\eta) = \tag{1}$$

$$\prod_{t=1}^{T} p(\Theta_t|\boldsymbol{\alpha}) \prod_{i=1}^{K} p(\beta_k|\boldsymbol{\eta}) \left(\prod_{n=1}^{N} P(w_{t,n}|z_{t,n},\beta_k) P(z_{t,n}|\Theta_t) \right) \tag{2}$$

2.2 Gibbs 采样

为了对 LDA 模型进行参数估计, 我们可以采用 Gibbs 采样的方法.

2.2.1 Gibbs 采样算法

Gibbs 采样可以认为是 Metropolis-Hasting 算法的一个特例. 其算法原理如下:

Algorithm 1 GIBBS 采样

初始化 $\{z_i : i = 1, \cdots, M\}$

for $\tau = 1, \cdots, T$ **do**

  Sample $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, \cdots, z_M^{(\tau)})$

  Sample $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, \cdots, z_M^{(\tau)})$

  $\vdots$

  Sample $z_j^{(\tau+1)} \sim p(z_j|z_1^{(\tau+1)}, \cdots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \cdots, z_M^{(\tau)})$

  $\vdots$

  Sample $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, \cdots, z_{M-1}^{(\tau+1)})$

2.2.2 估计 LDA 模型参数

这一步的过程是

- 通过对隐变量 θ 和 β 积分, 得到边缘概率 $P(z_d|w_d,\alpha,\eta)$.
  由于 $P(z|w,\alpha,\eta)=\frac{P(z|w,\alpha,\eta)}{P(w|\alpha,\eta)} \propto P(z,w,\alpha,\eta)=P(w|z,\alpha,\eta)P(z|\alpha,\eta)=P(w|z,\eta)P(z|\alpha)$, 分别考虑如下积分:

$$P(w|z,\eta)=\int p(w|z,\beta)p(\beta|\eta)d\beta=\prod_{k=1}^{K}\frac{B(\eta+n_k)}{B(\eta)} \tag{3}$$

$$p(z|\alpha)=\int p(z|\theta)p(\theta|\alpha)d\theta=\prod_{d=1}^{D}\frac{B(\alpha+n_d)}{B(\alpha)} \tag{4}$$

  其中, n_k 是一个 V 维向量, n_{kv} 表示数据中第 k 个话题生成第 v 个单词的次数; n_d 是一个 K 维向量, n_{dk} 表示第 d 个文档生成第 k 个主题的次数.

- 对后验概率进行吉布斯采样, 得到分布 $P(z_d|w_d,\alpha,\eta)$ 的样本集合. 对于吉布斯采样需要的条件概率, 可以依下式求出:

$$P(z_i=k'|z_{-i},\alpha,\eta)\propto\frac{P(z|w,\alpha,\eta)}{P(z_{-i}|w,\alpha,\eta)} \tag{5}$$

$$=\frac{\eta_{v'}+n_{k'v'}}{\sum_v\eta_v+n_{k'v}}\cdot\frac{\alpha_{k'}+n_{d'k'}}{\sum_k\alpha_k+n_{d'k}} \tag{6}$$

- 估计参数 θ_d 和 β_k 类似地有

$$p(\theta_d|z_d,\alpha)\propto\prod_{k=1}^{K}\theta_{dk}^{\alpha_k-1+n_{dk}}\Rightarrow p(\theta_d|z_d,\alpha)=Dir(\theta_d|n_d+\alpha)\Rightarrow\theta_{dk}=\frac{n_{dk}+\alpha_k}{\sum_k(n_{dk}+\alpha_k)} \tag{7}$$

  同理:

$$\beta_{kv}=\frac{n_{kv}+\eta_v}{\sum_v(n_{kv}+\eta_v)} \tag{8}$$

- 利用这个样本集合对参数 α 和 η 进行参数估计.

整体的算法可以按如下进行:

Algorithm 2 LDA 参数估计

1: 为每篇文档的每个词指派一个主题
2: **while** 未收敛 **do**
3: 通过 Gibbs 采样对每个 z_{dv} 进行估计.
4: 根据估计的结果计算 n_{kv} 和 n_{dk}, 进而估计参数 θ_d 和 β_k
5: **return** θ_d 和 β_k

2.2.3 训练终止条件/收敛条件

显然, 两次抽样结果(z)的差异能够反应整个训练过程是否趋于稳定. 如果连续两次Δz 差异不大, 则说明训练已经趋于收敛:

$$Delta\_Error=\frac{|last\_z-z|}{\sum_{doc}doc的词数}<eps$$

3 程序概要

3.1 数据集介绍

数据集有

- 助教给出的 **text.npy** (一个新闻数据集), 其中包含有 20 个主题, 共有 1000 条文档.
- 为了验证模型正确性自行构造的 **cat-computer** 数据集, 其中包含 8 条关于猫这种动物的描述, 6 条关于计算机及其部件的描述. 这些描述主要摘录自 wikipedia. 总共是两个主题, 即**猫**和**计算机**.

3.2 预处理

首先, 考虑到数据集中, 有的文本只包含一些符号, 有的则甚至是空文档. 为了过滤这些, 我调用了 `langdetect.detect()` 函数来判断文本语言, 从而只保留英文文本.

然后, 又借由 `sklearn.feature_extraction.text.CountVectorizer` 来提取字典, 而后通过以下规则进行进一步的过滤:

- 借助语料库 `nltk.corpus.brown.tagged_words()` 尽可能只保留名词
- 去除标点符号
- 通过 `nltk.corpus.stopwords` 过滤停用词
- 根据一系列正则表达式过滤特殊词

```
if word in brown_taged and brown_taged[word] != 'NN':
    word = ''
```

图 2: 尽可能保留名词

```
punctuations = {'(', ')', '{', '}', '[', ']', '"', "'",
                ',', ';', '.', '!', '?'}
```

图 3: 去除的标点符号

```
stopwords_set = set(stopwords.words('english'))
```

图 4: 过滤停用词

```
exception_search_pattern = re.compile(r'\^+|=+|~+|-\|*-|_+|\|_*\|')
exception_match_pattern = re.compile(r'^[\']?[-+]?([0-9]+(\.[0-9]+)?|\.[0-9]+)|(/\\)+|hm+')
should_search_pattern = re.compile(r'[\w]+')
```

图 5: 根据一系列正则表达式过滤

经过这些预处理, 整个词库量缩减了很多, 如下表所示:

	原文档	sklearn 预处理	进一步过滤
新闻数据集词汇量	28323	19285	11206
cat-computer 数据集	804	745	250

最后能够得到一个矩阵表示的文档集, $doc\_vec \in N^{D \times V}$, 每个元素表示对应文档中对应单词出现次数.

3.3 参数设置

主要涉及到的参数就是 α, β, max_epoch 及话题数目. 如下:

	α	β	max_epoch	话题数
新闻数据集	均0.01	均0.01	1000	20
cat-computer 数据集	均0.01	均0.01	100	2

4 实验结果

由于还有自己做的一个小数据集 Cat-Computer 数据集, 这里先以定性的角度分析一下该数据集的结果, 然后在下一部分中, 再定性+定量地分析一下新闻数据集的结果.

4.1 Cat-Computer 数据集

```
for i in range(topic_keywords.shape[0]):
    print(f'{i}: ', end='')
    for j in range(topic_keywords.shape[1]):
        print(topic_keywords[i][j], end=', ')
    print()
```

```
0: cat, human, frequency, feral, smell, species, night, communication, prey, house,
1: computer, use, memory, hardware, data, output, graphics, unit, gpus, circuit,
```

图 6: Cat-Computer 数据集实验结果

可以看到这里面两个话题的关键词都非常的准确, 具体而言:

- Cat 类话题关键词: 猫, 人, 野生的, 嗅觉, 物种, 夜晚, 交流, 猎物 等词汇都能很好地体现这个话题
- Computer 类话题关键词: 计算机, 内存, 硬件, 数据, 输出, 图形的, 单元, GPU, 电路等词汇都完美地体现了计算机这一主题.

在小数据集上, 可以发现 LDA 模型表现非常的优秀, 因此我们基本可以断定, 这个程序是能够正常运行, 并给出对应话题及其相关的关键词的. 于是我们就可以进一步探究本实验给出的官方数据集——新闻数据集.

4.2 新闻数据集

4.2.1 训练过程曲线

整个训练过程中, 为了观察模型是否收敛, 我们优先考虑了连续两次模型预测结果的差异, 同时也参考 θ 和 β 的变化. 经过统计及可视化, 可以得到如下图表:

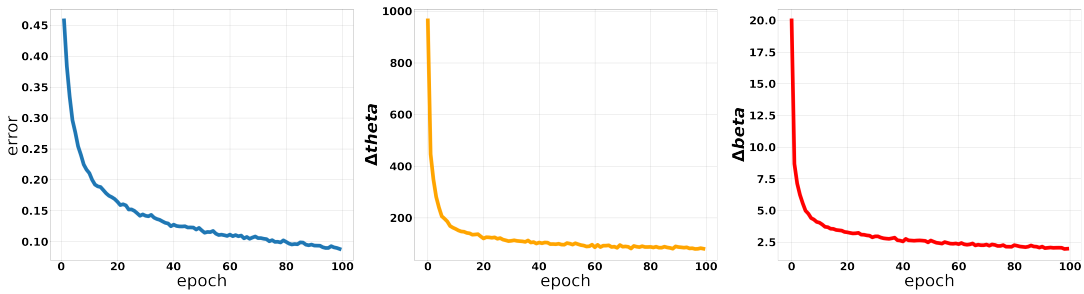

图 7: 新闻数据集训练过程, 左图为连续两次模型预测结果的差异, 中图为 θ 的变化, 右图为 β 的变化

很显然, 这三个值都呈指数级下降的趋势, 说明我们的训练过程是非常有效的. 值得注意的是, 最终 error 低达 0.10, 这意味着两次之间的词汇到话题的预测只有 0.10 的差异. 而反映了文档中话题的分布情况的 θ 及反映话题中关键词分布情况的 β 也都趋于稳定. 其中, 值得说明的是:

- θ 差异趋于 80 左右, 这说明 1000 篇文档的话题分布中, 只有 8% 的话题分布不太稳定
- β 差异趋于 2.0 左右, 这说明 20 个话题中只有 2% 的关键词分布不太稳定

4.2.2 话题的关键词结果说明

下图是 news 数据集的训练结果, 每一行是 10 个关键词, 表示了该话题的 top10 关键词.

```
0: time, reference, fact, science, place, watch, cause, interest, term, year,
1: jesus, god, time, year, life, day, book, sense, christian, thought,
2: time, use, hand, car, post, case, support, problem, light, thing,
3: government, state, system, idea, use, bit, way, control, thing, power,
4: part, world, man, time, god, reason, use, matter, fact, freedom,
5: contact, source, center, middle, guy, price, article, condition, sale, army,
6: chip, etc, clipper, use, law, thing, person, fact, government, area,
7: edu, time, mail, info, use, thing, card, cost, system, note,
8: software, system, bit, support, application, problem, paper, machine, printer, macintosh,
9: place, way, time, product, order, waste, etc, use, michael, computer,
10: man, course, support, time, city, thought, problem, food, act, death,
11: use, problem, line, file, information, email, code, time, dos, speed,
12: problem, way, etc, couple, information, mail, case, reply, list, canada,
13: space, year, data, shuttle, thought, number, light, research, use, program,
14: person, question, human, way, year, return, day, morning, second, case,
15: way, chance, memory, day, average, cause, rate, course, use, defense,
16: edu, program, ftp, version, pub, name, screen, space, machine, stuff,
17: system, etc, way, area, problem, turn, set, control, hand, test,
18: edu, pitt, gordon, geb, card, money, year, surrender, cadre, n3jxp,
19: thing, play, bike, hell, game, dod, nhl, team, head, mistake,
```

图 8: 新闻数据集结果

可以看到, 里面虽然有部分话题的含义比较模糊不清, 但有一些比较突出的值得说明一下:

- 0号话题中的: 时间, 引用, 事实, 科学, 原因等词汇能够反映**科学**这一话题
- 1号话题中的: 耶稣, 上帝, 生命, 基督, 书籍等词汇能够反映**宗教信仰**这一话题
- 3号话题中的: 政府, 国家, 制度, 理念, 控制, 权力等词汇能够反映**政治**这一话题
- 7号话题中的: edu, 时间, 邮件, 信息等词汇能够反映**邮件/电子邮件**这一文本背景. 识别出这样的东西是很合理的, 因为数据集中出现了大量的邮件文本.
- 8号话题中的: 软件, 系统, 比特, 支持, 应用, 问题, 论文, 机器, 打印机, Macintosh(苹果公司早年电脑型号) 就完美地适配了**电脑/计算机**这一主题
- 9号话题中的: 产品, 订单, 位置等词汇能够一定程度地体现**商业**话题
- 13号话题中的: 太空, 年, 数据, 航天飞机, 思考, 数字, 光, 研究, 使用, 计划等词汇则可以反映**航天**这一话题
- 19号话题中的: 事件, 运动, 自行车, 比赛, 国家冰球联盟, 团队, 领导, 错误等词汇则可以反映**体育赛事**

4.2.3 文档分类结果说明

通过以下代码我们可以找出, 每个话题中最具代表性的文本. 由于 theta 每行表示对应文档的话题分布, 因此找 theta 每列最值的索引, 可以给出每个话题最具代表性的文本. 其中输出结果第一行是每个话题对应最具代表性的文本, 比如 687 是话题 1 最具代表性的文本, 而第二三行是其对应概率, 这里为 0.916.

```
n_word, n_topic = lda.count(dataset, lda.z)
theta = lda.get_theta(dataset, n_topic)
print(np.argsort(-theta, axis=0)[:1])
print(np.max(theta, axis=0))
```

```
[[420 687 127 626 134 155 277 572 301 755 140 121 692 339 237 794 761 760 511 433]]
[0.99349315 0.91619718 0.99059406 0.96346154 0.99214876 0.96935484 0.99059406 0.97682927 0.97934783 0.96935484
 0.96935484 0.98442623 0.98137255 0.940625   0.99274809 0.96935484 0.98303571 0.98560606 0.9875     0.99246032]
```

图 9: 获取每个话题中最具代表性的文本

下面举两个例子说明:

1. 例子1: 上一节中, 我们提到 1 号话题与宗教信仰关联, 输出其对应最具代表性的文档, 会发现它是关于”启示录”中12:7-9的内容, 见图10.

2. 例子2: 话题19对应的是体育赛事, 可以发现对应最具代表性的文档如下, 是一个体育赛事的记分榜之类的文本, 见图11.

```
The quick answer:  Revelation 12:7-9

"And there was war in heaven.  Michael and his angels fought against
the dragon and his angels who fought back.  But he [the dragon] was
not strong enough, and they lost their place in heaven.  The great
dragon was hurled down--that ancient serpent, called the devil and
Satan, who leads the whole world astray.  He was hurled down to the
earth, and his angels with him."

The earlier part of chapter 12 deals (very symbolically) with why
Satan rose up in battle against Michael and the good angels in the
first place.
```

图 10: 最可能是1号话题(宗教信仰话题)的文本

```
Group A                          Group B
~~~~~~~                          ~~~~~~~

Cardiff Devils  7-3  Bracknell Bees     Nottingham Panthers  8-3  Billingham
Humberside      7-7  Whitley Warriors   Murrayfield Racers  11-2  Fife Flyers
Whitley Bay     6-9  Cardiff Devils     Billingham Bombers   6-8  Murrayfield
Humberside      8-5  Bracknell Bees     Nottingham Panthers 11-5  Fife Flyers
Cardiff Devils 10-4  Humberside         Murrayfield Racers   6-4  Nottingham
Bracknell Bees  4-9  Whitley Bay        Fife Flyers          2-5  Billingham
Bracknell Bees  3-8  Cardiff Devils     Billingham Bombers   2-8  Nottingham
Whitley Bay     5-7  Humberside         Fife Flyers          3-12 Murrayfield

                P W D L  F  A  P                          P W D L  F  A  P
Cardiff Devils  4 4 0 0 34 16  8*       Murrayfield Racers   4 4 0 0 37 15  8*
Humberside      4 2 1 1 26 27  5        Nottingham Panthers  4 3 0 1 31 16  6*
Whitley Bay     4 1 1 2 27 27  3        Billingham Bombers   4 1 0 3 16 26  2
Bracknell Bees  4 0 0 4 15 32  0        Fife Flyers          4 0 0 4 12 39  0

* indicates qualified for Championship Finals

Relegation/Promotion A           Relegation/Promotion B
~~~~~~~~~~~~~~~~~~~~~~           ~~~~~~~~~~~~~~~~~~~~~~

Basingstoke    10-4  Swindon Wildcats   Sheffield Steelers  12-8  Peterborough
Durham Wasps   13-5  Romford Raiders    Slough Jets          1-9  MK Kings
Basingstoke     6-0  Durham Wasps       Sheffield Steelers   9-4  Milton Keynes
Swindon         8-5  Romford Raiders    Milton Keynes Kings  4-6  Peterborough
Durham Wasps   17-2  Swindon Wildcats   Slough Jets          2-12 Sheffield
Romford         4-10 Basingstoke        Peterborough        10-2  Slough Jets
Romford        *8-3* Durham Wasps       Peterborough         8-5  Sheffield
Swindon         7-11 Basingstoke        Milton Keynes Kings 10-4  Slough Jets

                P W D L  F  A  P                          P W D L  F  A  P
Basingstoke     4 4 0 0 37 15  8        Sheffield Steelers   4 3 0 0 38 22  6
Durham Wasps    4 2 0 2 33 21  4        Peterborough         4 3 0 1 32 23  6
Swindon         4 1 0 3 21 43  2        Milton Keynes Kings  4 2 0 2 27 20  4
Romford Raiders 4 1 0 3 22 34  2        Slough Jets          4 0 0 4  9 41  0
```

图 11: 最可能是19号话题(体育赛事话题)的文本

5 实验总结

本次 LDA 实验通过一个新闻数据集, 让我深刻理解了 LDA 算法原理与流程. 令我震惊的是, 这个话题模型甚至能够识别出记分榜这样的数据, 这说明本次实验完成的模型是非常有效的.