机器学习实验

李晨亮
武汉大学国家网络安全学院

任务1 文本分类

➢ 数据集

- 20NG 文本分类数据集 - http://qwone.com/~jason/20Newsgroups/ (使用bydate版本，包含已经分好的训练集与测试集)
- 一共20个类 / 验证集可从训练集中按一定比例(自行确定)采样构建

➢ 任务说明

- 通过编程实现基于感知机、~~K近邻、~~朴素贝叶斯模型的文本分类模型
  - 评价指标: Macro-averaging F1 (见参考论文" A Class-Feature-Centroid Classifier for Text Categorization")
- ~~通过使用已有SVM工具，通过设计文档的向量表示，实现文本分类 (见参考论文" A Class-Feature-Centroid Classifier for Text Categorization"和"其他参考资料")~~
  - 评价指标: Macro-averaging F1

➢ 其他参考资料(推荐)

- https://blog.datasciencedojo.com/unfolding-naive-bayes-from-scratch-part-1/
- https://aiiseasy.com/2019/06/09/text-classification-svm-naive-bayes-python/
- https://sebastianraschka.com/Articles/2014_naive_bayes_1.html
- https://nlp.stanford.edu/IR-book/html/htmledition/text-classification-and-naive-bayes-1.html
- https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

任务1 文本分类

➢ 实验报告

- 数据集预处理与统计分析
- 各文本分类模型的实现设计说明
- 算法核心部分的代码实现
- 各文本分类模型的超参数的敏感性分析
- 各文本分类模型的最优设置下的实验结果对比
- 实验感想

任务1 文本分类

➢ 文本预处理

- 目标: 将每篇文档用N维特征向量进行表示

$x = (x_1, x_2, \cdots\cdots, x_N)$

- 如何选择N维特征?
  - 文本由其所包含的单词组成 → 词袋模型（Bag-of-Words）
  - N = 文档集中出现的单词数（去重）= |V| （V代表文档集的词典/词项集合）
- 文本预处理环节

- D_1: "Each state has its own laws."
- D_2: "Every country has its own culture."

V=
{*each*, *state*, *has*, *its*, *own*, *laws*,
every, *country*, *culture*}

Table 1. *Bag of words representation of two sample documents D_1 and D_2.*

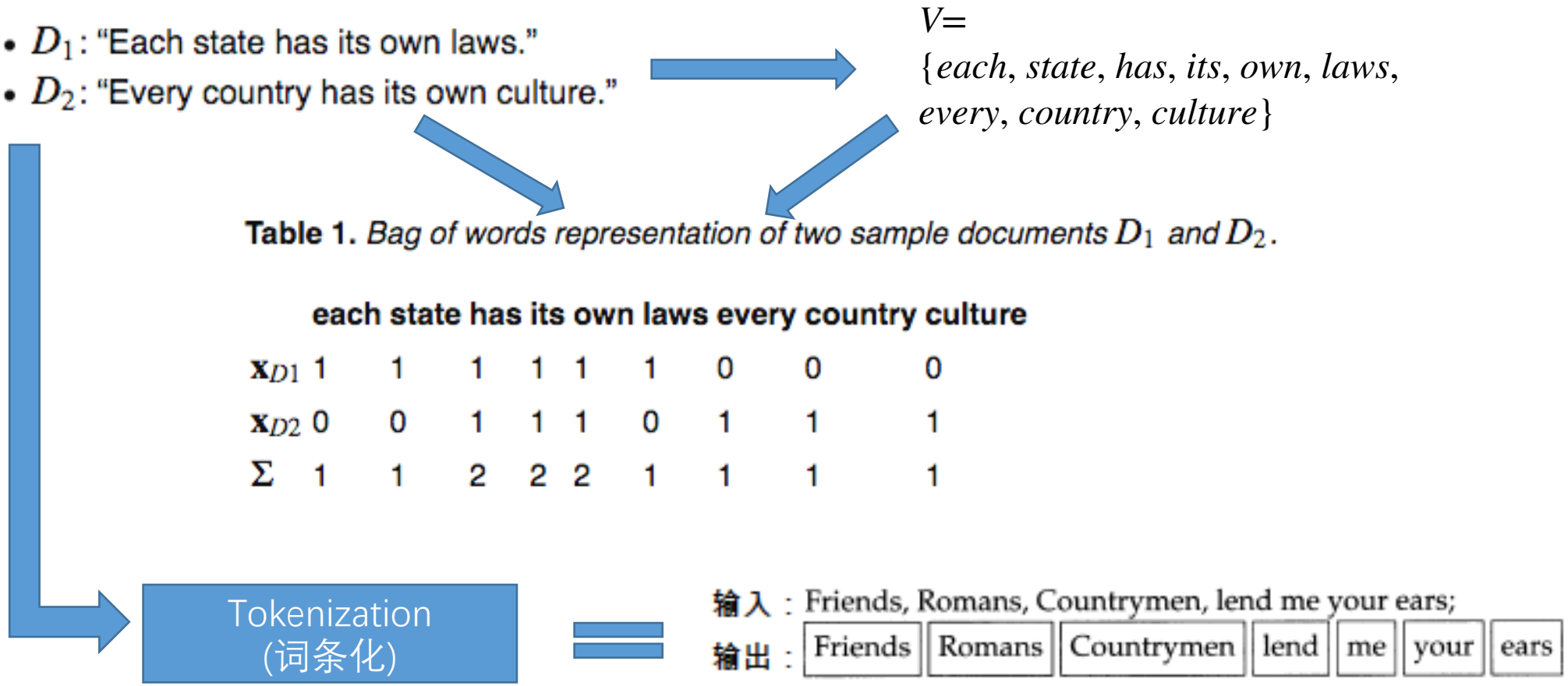
	each	state	has	its	own	laws	every	country	culture
$\mathbf{x}_{D1}$	1	1	1	1	1	1	0	0	0
$\mathbf{x}_{D2}$	0	0	1	1	1	0	1	1	1
Σ	1	1	2	2	2	1	1	1	1

维度对应单词出现在文档中的**次数**

任务1 文本分类

➢ 文本预处理

- 文本预处理环节



任务1 文本分类

➢ 文本预处理

- 文本预处理环节

Tokenization
(词条化)

输入：Friends, Romans, Countrymen, lend me your ears;

输出：

Friends	Romans	Countrymen	lend	me	your	ears

Table 2. *Example of tokenization.*

A swimmer likes swimming, thus he swims.						
↓						
a	swimmer	likes	swimming	thus	he	swims

1. 按照空格、回车先切分字符串 → 原始词条
2. 原始词条 → 去掉标点符号
3. 大写字符转换为小写字符
4. **去掉属于停用词的词条**

Table 3. *Example of stop word removal.*

A swimmer likes swimming, thus he swims.					
↓					
swimmer	likes	swimming	,	swims	.

任务1 文本分类

➢ 文本预处理

- 文本预处理环节

Table 1. *Bag of words representation of two sample documents D_1 and D_2.*

	each	state	has	its	own	laws	every	country	culture
$\mathbf{x}_{D1}$	1	1	1	1	1	1	0	0	0
$\mathbf{x}_{D2}$	0	0	1	1	1	0	1	1	1
Σ	1	1	2	2	2	1	1	1	1

1. 文档的长度有限
2. 文档包含的词项有限
3. 文档集的词典较大（N非常大）

$x = (x_1, x_2, \cdots\cdots, x_N)$ 包含太多的0值占用太多存储空间，增加无谓的计算量

稀疏编码

任务1 文本分类

➢ 文本预处理

- 文本预处理环节

Table 1. *Bag of words representation of two sample documents D_1 and D_2.*

	each	state	has	its	own	laws	every	country	culture
$\mathbf{x}_{D1}$	1	1	1	1	1	1	0	0	0
$\mathbf{x}_{D2}$	0	0	1	1	1	0	1	1	1
Σ	1	1	2	2	2	1	1	1	1

1. 文档的长度有限
2. 文档包含的词项有限
3. 文档集的词典较大（N非常大）

$x = (x_1, x_2, \cdots\cdots, x_N)$ 包含太多的0值占用太多存储空间，增加无谓的计算量

稀疏编码

任务1 文本分类

➢ 文本预处理

- 文本预处理流程

1 构建文档集的词典

词条化

词典文件

```
1  each
2  state
3  has
4  its
5  own
6  laws
7  every
8  country
9  culture
```

HashMap<String, Integer>
<词项, ID>

2 构建文档的稀疏表示

- D_1: "Each state has its own laws."

词条化

稀疏表示

<词项ID, 次数>
{<1,1>, <2,1>, <3,1>, <4,1>, <5,1>, <6,1>}

写入本地文件: "文档ID: <1,1>, <2,1>, <3,1>, <4,1>, <5,1>, <6,1>"

任务1 文本分类

➢ 感知机

- 每个类有一个N维的w向量 + 偏置b
- 一次Load进一个文档的稀疏表示或K个文档的稀疏表示（**K量力设置**）
  - 根据梯度下降调整每个类的参数（w向量+偏置b）

$$f(x) = \text{sign}(w \cdot x + b)$$

- 称为感知机，
- 模型参数：w x，内积，权值向量，偏置，
- 符号函数：

$$\text{sign}(x) = \begin{cases} +1, & x \geqslant 0 \\ -1, & x < 0 \end{cases}$$

- 文档的N维特征表示

➢ The *tf-idf* weight of a term is the product of its *tf* weight and its *idf* weight.

- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

$$w_{t,d} = (1 + \log_{10} \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

任务1 文本分类

➢ 朴素贝叶斯

- 每个类通过其训练文档的稀疏表示, 统计每个词项出现在属于该类文档的总次数
- 保存这个<词项, 总次数>到一个文档
- 保存每个类的先验概率到一个文档
- 测试时需要Load进这些文档中的统计量，通过**贝叶斯估计**进行分类