基于 Hadoop MapReduce 并行计算框架的邮件自动分类

曹佳涵　白家杨　刘笑今
2019st04 小组

摘　要

本实验的目标是通过 MapReduce 编程来实现邮件的自动分类，通过本课程设计的学习，可以体会如何使用 MapReduce 完成一个综合性的数据挖掘任务，包括全流程的数据预处理、样本分类、样本预测等。我们使用 Hadoop MapReduce 并行计算框架对原始邮件文本进行特征选择、特征向量权重计算、文本分类和样本预测任务，从而搭建起一个完整的文本分类处理流程。我们采用了不同的特征提取方法和分类算法并进行比较和分析。此外，我们使用 Spark 同样完成了实验。

关键词： 邮件分类，并行计算，MapReduce，Spark

1　引入

此模板是基于 LaTeX 的标准文类 article 设计，也即意味着你可以把 article 文类的选项传递给本模板，比如 `a4paper`，`10pt` 等等（推荐使用 `11pt`）。本模板支持 PDFLaTeX 和 XeLaTeX[1] 两种编译方式。

数学字体的效果如下：

$$(a+3b)^n = \sum_{k=0}^{n} C_n^k a^{n-k}(3b)^k \tag{1}$$

1.1　全局选项

我在这个模板中定义了一个语言选项 `lang`，可以选择英文模式 `lang=en`（默认）或者中文模式 `lang=cn`。当选择中文模式时，图表的标题引导词以及参考文献，定理引导词等信息会变成中文。你可以通过下面两种方式来选择语言模式：

```
\documentclass[lang=cn]{elegantpaper} % or
\documentclass{cn}{elegantpaper}
```

[1] 中文字体均使用 ctex 包设置。

1.2 自定义命令

在此模板中，并没有修改任何默认的命令或者环境，所以，你可以在此模板使用原来的命令和环境。另外，我自定义了 3 个命令：

1. \email：创建邮箱地址的链接；
2. \figref：用法和 \ref 类似，但是会在插图的标题前添加 < **图 n**> ；
3. \tabref：用法和 \ref 类似，但是会在表格的标题前添加 < **表 n**>；
4. \keywords：为摘要环境添加关键词。

1.3 列表环境

你可以使用列表环境（itemize、enumerate、description），示例如下：

```
\begin{itemize}
   \item Routing and resource discovery;
   \item Resilient and scalable networks;
   \item Distributed storage and search.
\end{itemize}
```

- Routing and resource discovery;
- Resilient and scalable networks;
- Distributed storage and search.

1.4 插图

插图的命令和以前一样，也是使用 figure 环境。**图 1** 显示了插图的效果。你可以把你的图放到当前工作目录的如下子目录下 (./image/, ./img/, ./figure/, ./fig/)。

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.6\textwidth]{scatter.pdf}
  \caption{Scatter Plot Example \label{fig:scatter}}
\end{figure}
```

1.5 表格

我强烈建议你使用 booktabs 宏包，这个宏包有三个命令 \toprule、\midrule 和 \bottomrule 能方便你制作三线表。**表 1** 是一个示例：

```
\begin{table}[htbp]
  \small
  \centering
  \caption{Auto MPG and Price \label{tab:reg}}
    \begin{tabular}{lcc}
    \toprule
                    &       (1)          &         (2)       \\
    \midrule
```

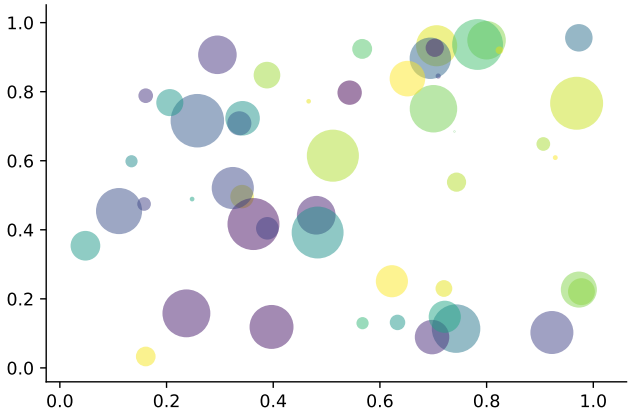

图 1: Scatter Plot Example

```
    mpg             &     -238.90***      &       -49.51      \\
                    &      (53.08)        &       (86.16)     \\
    weight          &                     &       1.75***     \\
                    &                     &       (0.641)     \\
    constant        &      11,253***      &        1,946      \\
                    &      (1,171)        &       (3,597)     \\
    obs             &         74          &          74       \\
    $R^2$           &       0.220         &       0.293       \\
    \bottomrule
    \multicolumn{3}{l}{\scriptsize Standard errors in parentheses} \\
    \multicolumn{3}{l}{\scriptsize *** p<0.01, ** p<0.05, * p<0.1} \\
    \end{tabular}%
\end{table}%
```

表 1: Auto MPG and Price

	(1)	(2)
mpg	-238.90***	-49.51
	(53.08)	(86.16)
weight		1.75***
		(0.641)
constant	11,253***	1,946
	(1,171)	(3,597)
obs	74	74
R^2	0.220	0.293

Standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

1.6 参考文献

此模板使用了 BibTEX 来生成参考文献，默认使用的文献样式（bib style）是 GB/T 7714-2015[2]。参考文献示例：？使用了中国一个大型的 P2P 平台（人人贷）的数据来检验男性投资者和女性投资者在投资表现上是否有显著差异。

你可以在谷歌学术，Mendeley，Endnote 中获得文献条目（bib item），然后把它们添加到 wpref.bib 中。在文中引用的时候，引用它们的键值（bib key）即可。注意需要在编译的过程中添加 BibTEX 编译。如果你想在参考文献中添加未引用的文献（部分或者全部），可以使用

```
\nocite{EINAV2010, Havrylchyk2018} % add the two reference.
\nocite{*}   % add all the reference in the bib file.
```

如果你想修改参考文献的样式（比如改为 aer），你可以在导言区将下面代码注释掉。

```
\usepackage[authoryear]{gbt7714}
```

并且文档末尾添加

```
\bibliographystyle{aer}
```

2 数据分析

共有 20 个类别，20000 个文件.....

3 MapReduce 处理流程

3.1 特征选择

本任务的主要工作是对原始的邮件文本中进行特征选择，选择出能够表征邮件主题的特征词，为后续的文本分类做准备。

对于输入的未分词的邮件训练样本全集和停词表，我们需要输出全局邮件文本特征，并对它们进行相应的编号，此外，对于训练数据集的目录，将目录名（即文本类别）转换为相应的类别序号。

对于该步骤，我们采用了两种计算方法，分别为半并行化和并行化计算方法。

3.1.1 半并行化计算方法

半并行化计算方法是指将该任务划分为两个步骤，第一步读取训练样本全集和停词表，生成干净文本；第二步读取干净文本，生成全局邮件特征集。

第一步的主要思路是顺序执行，首先读取 20_newsgroup 文件夹下的子文件夹，将子文件夹名（即类别名）转化为类编号并进行存储。将 Lucene 的 Standard Analyzer 分词器作为基类，输入停词

[2]通过调用 gbt7714 宏包

表，自定义自己的带有停词表功能的停词器，然后顺序依次读取每个子文件夹下的所有文件，使用自定义的停词器对文本进行分词，并将分词结果储存在目标文件中，我们将分词后的文本称为干净文本，生成干净文本的过程是顺序执行，非并行化的。为了后续处理的方便性，我们修改输出文件的格式为：filename-classNum，filename 为原文件的文件名，classNum 为它所属的类别的编号，两者用分隔符'-' 隔开。

伪代码如下：第二步从干净文本中提取全局邮件文本特征集的过程是并行化的，使用 MapReduce

Algorithm 1 特征选择半并行化算法：第一步

Input: 邮件训练样本全集 U，停词表 S

Output: 用停词表分割后的干净文本

```
初始化类别名-类别编号对应表 classMap
初始化停词器 stopAnalyzer=StopAnalyzer(S)
for each 文本 u in U do
    新文本 u′ = stopAnalyzer(u)
    通过类别名-编号对应表查找文本 u 的文件名 filename 对应的类别编号 num=classMap(u)
    在目标目录下创建 filename-num 文件，并将新文本 u′ 的内容写入该文件。
end for
```

并行计算框架，在 Map 阶段读取干净文本，对于每一个单词 word，发射 (word,1) 键值对。在 Reduce 阶段，设置一个全局缓存变量 n，为每个单词维护一个编号，并将每个单词和相应的编号输出到全局邮件特征集中。

伪代码如下：

3.1.2 并行化计算方法

非并行化顺序执行，缺点也显而易见：计算速度慢，效率低，因此我们设计了并行化的计算方法，利用 MapReduce 计算框架，很好地并行处理大量的训练文本，极大地加快了处理速度。

在 Map 阶段读取原文本，利用停词器将其分词后输出到目标目录下，同时对于每一个单词，发射 (word,filename-classNum) 键值对。值得注意的是，因为停词表是所有 Mapper 都需要各自初始化的，因此将其初始化放在 Map 阶段的 setup 过程中，首先将停词表文件路径存在 DistributedCache 中，然后在 setup 过程中读取该路径并利用其初始化停词器，在 map 过程中使用该停词器。

在 Reduce 阶段，设置一个全局缓存变量 n，用来表示每一个单词的唯一标号。输入为 (word, value)。由于在 Reduce 阶段，已经将相同 key 的键值对都整合到了一起，因此读取的 word 值是唯一不重复的，只需要利用全局变量 n 为每一个单词分配相应的标号，并将单词和编号输出到目标文件中即可。

伪代码如下：

Algorithm 2 特征选择半并行化算法：第二步

Input: 干净文本 U'

Output:

Map 阶段：
function MAP($filename$, $text$)
    for each word w in $text$ **do**
        Emit(w,1)
    end for
end function
Reduce 阶段：
function REDUCE($word$, $value$)
    全局缓存变量 $num = num + 1$
    将 $word$ 和 num 写入到全局邮件文本特征集中
    Emit($word$,$value$)
end function

3.1.3 运行结果

该任务的输出结果如下图所示：

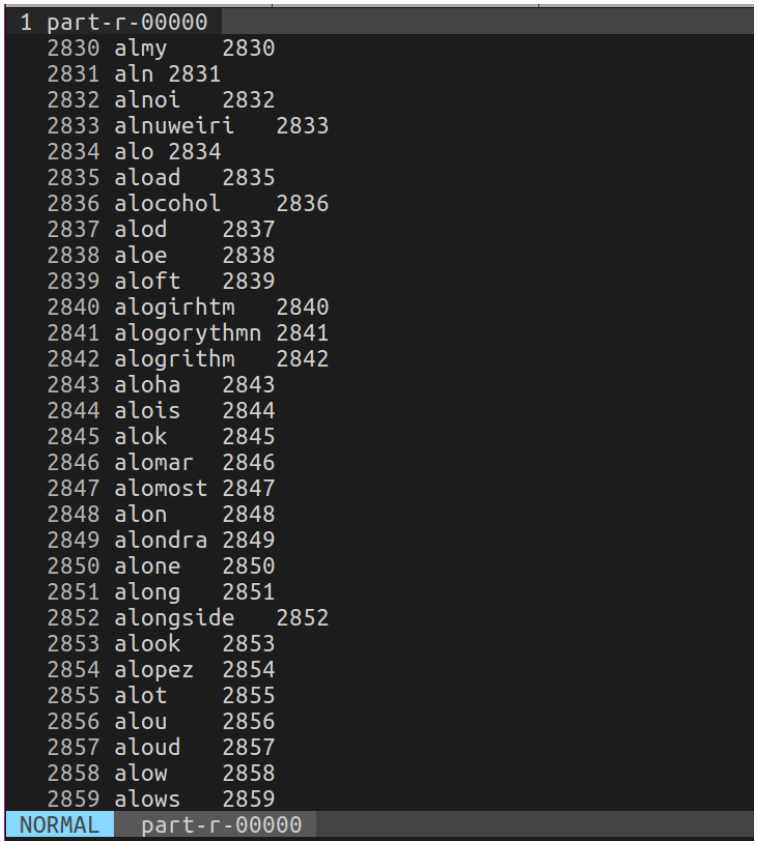
1 part-r-00000
2830 almy 2830
2831 aln 2831
2832 alnoi 2832
2833 alnuweiri 2833
2834 alo 2834
2835 aload 2835
2836 alocohol 2836
2837 alod 2837
2838 aloe 2838
2839 aloft 2839
2840 alogirhtm 2840
2841 alogorythmn 2841
2842 alogrithm 2842
2843 aloha 2843
2844 alois 2844
2845 alok 2845
2846 alomar 2846
2847 alomost 2847
2848 alon 2848
2849 alondra 2849
2850 alone 2850
2851 along 2851
2852 alongside 2852
2853 alook 2853
2854 alopez 2854
2855 alot 2855
2856 alou 2856
2857 aloud 2857
2858 alow 2858
2859 alows 2859
NORMAL part-r-00000

图 2: 全局邮件文本特征

Algorithm 3 特征选择并行化算法

Input: 邮件训练样本全集 U，停词表 S

Output: 用停词表分割后的干净文本，全局邮件文本特征集

初始化类别名-类别编号对应表 classMap
将停词表 S 存入 DistributedCache 中
Map 阶段：
function SETUP
    从 DistributedCache 中读取停词表 S
    初始化停词器 stopAnalyzer=StopAnalyzer(S)
end function
function MAP($filename$, $text$)
    $text'$ = stopAnalyzer($text$)
    将 $text'$ 写入到目标干净文本中
    for each word w in $text'$ **do**
        Emit(w,$filename$)
    end for
end function
Reduce 阶段：
function REDUCE($word$, $value$)
    全局缓存变量 $num = num + 1$
    将 $word$ 和 num 写入到全局邮件文本特征集中
    Emit($word$,$filename$)
end function