计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210400280

基于 Spark 的舆情情感大数据分析集成方法

戴宏亮[1]　钟国金[1]　游志铭[1]　戴宏明[2]
1 广州大学经济与统计学院　广州 510006
2 华南理工大学软件学院　广州 510006
(hldai618@gzhu.edu.cn)

摘　要　随着移动互联技术的不断发展,社交媒体成为了公众分享观点和抒发情感的主要平台,在重大社会事件下对社交媒体文本进行情感分析能够有效监控舆情。针对现有中文社交媒体情感分析算法的准确性能和运行效率较低的问题,提出了一种基于 Spark 分布式系统的集成情感大数据分析方法(Spark Feature Weighted Stacking,S-FWS)。该方法首先基于 Jieba 库预分词和 PMI 关联度完成新词发现;然后考虑词语重要度混合提取文本特征,并使用 Lasso 进行特征选择;最后改进传统 Stacking 框架忽略特征重要度的缺点,使用初级学习器的准确率信息对类概率特征进行加权处理并构造多项式特征,进而训练次级学习器。分别在单机模式和 Spark 平台下引入多种算法进行对比实验,实验结果证明所提 S-FWS 方法的准确性能和耗时性能具备一定优势,并且分布式系统能够大幅提高算法的运行效率,同时随着集群工作节点的增加,算法耗时逐渐降低。

关键词:情感分析;舆情;中文社交媒体;Spark;Stacking
中图法分类号　TP391

Public Opinion Sentiment Big Data Analysis Ensemble Method Based on Spark

DAI Hong-liang[1],ZHONG Guo-jin[1],YOU Zhi-ming[1] and DAI Hong-ming[2]
1 School of Economics and Statistics,Guangzhou University,Guangzhou 510006,China
2 School of Software,South China University of Technology,Guangzhou 510006,China

Abstract　With the development of mobile Internet technology,social media has become the main approach for the public to share views and express their emotions. Sentiment analysis for social media texts in major social events can effectively monitor public opinion. In order to solve the problem of low accuracy and efficiency of existing Chinese social media sentiment analysis algorithms,an ensemble sentiment analysis big data method(S-FWS) based on Spark distributed system is proposed. Firstly,the new words are found by calculating the PMI association degree after pre-segmentation by Jieba library. Then,the text features are extracted by considering the importance of words and feature selection is realized by Lasso. Finally,in order to improve the traditional Stacking framework neglecting the feature importance,the accuracy information of the primary learners is used to weight the probabilistic features,and the polynomial features are constructed to train the secondary learner. A variety of algorithms are introduced in the stand-alone mode and the Spark platform receptively to carry out comparative experiments. Results show that the S-FWS method proposed in this paper has certain advantages in accuracy and time consumption;distributed system can greatly improve the operating efficiency of the algorithms,and with the increase of working nodes,the time consumption of the algorithms gradually decreases.

Keywords　Sentiment analysis,Public opinion,Chinese social media,Spark,Stacking

1　引言

随着中文社交媒体的发展,重大事件和社会争议的发生能够迅速产生海量的文本数据,学术界和工业界对舆情事件情感分析[1]的重视程度越来越高。情感分析是通过运用自然语言处理、机器学习等相关技术研究文本中蕴含的主观情绪的新兴课题[2-4]。在中文社交媒体领域,文本存在语法结构不完整、网络新词过多等特点,这些特点不利于算法准确地完成文本分类任务,并且面对海量数据,算法运行效率低下,这为舆情监管带来了极大挑战。

到稿日期:2021-04-26　返修日期:2021-06-16
基金项目:国家社会科学基金项目(18BTJ029)
This work was supported by the National Social Science Foundation(18BTJ029).
通信作者:戴宏明(1355369191@qq.com)

首先，集成学习算法能够很好地解决文本分类任务中准确率低的问题。Wang 等[3]集成朴素贝叶斯和支持向量机等 6 种分类模型，对 Twitter 航空服务数据集进行情感分类，结果具有较高的准确度。Alrehili 等[5]利用 n-gram 模型提取英文文本特征，结合集成学习进行文本分类，实验结果表明集成模型效果优于简单模型。集成算法虽准确且性能优越，但复杂度高，运行耗时长。而以 Hadoop 和 Spark[6]为代表的分布式系统能够帮助集成算法高效地处理大规模数据，国内外学者对此进行了大量研究。Elzayady 等[7]利用 Spark 框架对 Twitter 文本数据进行情感分类，验证了分布式系统提升算法速度的有效性。Yang 等[8]提出了基于 Spark 平台的集成学习并行化算法，通过对不同数量级的文本进行实验，证明了该算法能够线性缩短文本分类时间，且可扩展性良好。Wang 等[9]提出了一种以 Spark 平台为基础，结合多种算法的集成算法来进行英文评论数据的情感分析，通过实验证明了该算法的耗时性能和准确性能良好。上述研究都对算法做出了一定的改进，但中文社交媒体文本区别于其他普通文本，其特殊性大大提高了算法的分类难度，需要在保证算法运行效率的同时兼顾准确性能。

针对以上问题，本文基于 Spark 分布式系统提出了一种应用于舆情情感分析的集成方法，并将其命名为 S-FWS。基于 3 个中文社交媒体文本数据集展开实验，结果表明 S-FWS 方法具备优秀的准确性能和较短的耗时，并且 Spark 系统大幅度提高了算法的运行效率。

2 相关技术

2.1 Spark 分布式运算平台

UC Berkeley 于 2009 年研发了一种基于内存运算的大数据计算平台——Spark[10-11]。Spark 由 Scala 编程实现，同时支持 Python 和 Java 编程语言等通过 API 接入完成开发任务。其继承了 MapReduce 容错性高和伸缩性强的优点，同时弥补了 MapReduce 必须严格执行先映射(Map)后规约(Reduce)的缺陷，不再采用 HDFS 迭代运算，而是通过有向无环图(Directed Acyclic Graph，DAG)算子传递中间结果到下一阶段任务。

Spark 以弹性分布式数据集(RDD)[12]为工作核心，通过 DAG 图和 Stage 作业划分完成组织、运算和调度等一系列计算任务，效率相比 MapReduce 有显著提升，其高效实现了大数据下的舆情情感分析、智能数据治理。

2.2 文本特征提取

2.1.1 点互信息算法

点互信息算法(Pointwise Mutual Information，PMI)[13]是一种常用的相关性度量算法，应用于文本领域，其通过计算两个给定词组的 PMI 值来衡量它们之间的关联程度，若 PMI 值越大，则表明这两个词组之间的相关性越强，具体计算式如下：

$$PMI(w_1,w_2)=\log_2\left(\frac{P(w_1,w_2)}{P(w_1)P(w_2)}\right) \tag{1}$$

其中，$P(w_1)$和 $P(w_2)$分别代表词组 w_1 和 w_2 在语料库中出现的概率，$P(w_1,w_2)$则是词组 w_1 和 w_2 在语料库中邻接出现的概率。基于大数定律，保证样本量较大时，能够将概率 $P(x)$替换为频率 $F(x)$，进而式(1)可以改写为：

$$PMI(w_1,w_2)=\log_2\left(\frac{F(w_1,w_2)N}{F(w_1)F(w_2)}\right) \tag{2}$$

2.2.2 TF-IDF 算法

TF-IDF 算法[14]是一种统计方法，其通过计算某特定词语在相应语料库中的词频 TF 和逆文本频率 IDF 的乘积来评估该词语的分类能力。某词语的 TF-IDF 值越大，代表其越重要，分类能力越强，具体计算式如下：

$$tf_{i,j}\times idf_i=\frac{n_{i,j}}{\sum_k n_{k,j}}\times\log\frac{|D|}{|\{j:t_i\in d_j\}|+1} \tag{3}$$

其中，分子 $n_{i,j}$是词 n_i 在该语料库 d_j 中出现的次数总和，分母 $\sum_k n_{k,j}$是该语料库 d_j 中所有的 k 个词出现的次数总和，$|D|$是该预料库中的文件总数，$|\{j:t_i\in d_j\}|$表示包含词语 t_i 的文件数目。

2.2.3 Word2vec 算法

Word2vec 算法[15]本质上是一种简单双层神经网络，主要包括 Skip-gram 模型和 CBOW 模型。Skip-gram 的算法流程是在神经网络中输入一个特定词的 One-Hot Encoder 词向量，通过训练最终输出该特定词所对应的上下文词的词向量，模型本质的数学表达式如下：

$$P(w_{(t-k)},w_{(t-k+1)},\cdots,w_{(t+k-1)},w_{(t+k)}\mid w_{(t)}) \tag{4}$$

其中，$w(t)$为文本中第 t 个词，k 为词语 $w(t)$上下文词语总数的一半。而 CBOW 模型的主要思想则相反：输入某特定词所对应上下文词的 One-Hot Encoder 词向量，通过训练得到该特定词的词向量，模型的本质数学表达式为：

$$P(w_{(t)}\mid w_{(t-k)},w_{(t-k+1)},\cdots,w_{(t+k-1)},w_{(t+k)}) \tag{5}$$

2.3 集成学习

2.3.1 Bagging

Bagging 是一种并行式集成方法，基于每个重采样子样本训练出若干个基学习器，利用各基学习器的独立性，对它们的预测结果求平均能够降低估计的方差。随机森林[16]是 Bagging 的代表，在分类任务中建立若干决策树为基学习器，将每一个决策树的结果通过投票法集成起来得到最终结果。随机森林在建立基学习器阶段增加了随机选择样本集和随机选择属性集两个要素，使得估计偏差得以降低。其简单的数学表达式如下：

$$F(x)=\frac{1}{M}\sum_{m=1}^{M}f_m(x) \tag{6}$$

2.3.2 Boosting

Boosting 是一种串行式集成方法，利用基学习器之间的强依赖性不断调整样本分布，通过加权结合降低估计的偏差。

梯度提升树(GBDT)[17]是 Boosting 的一种，以决策树为基函数，以加法模型为结合策略，使用梯度下降法优化目标函数。构建 M 个决策树，经过多次迭代将其最终组合为一个强分类器。每一次迭代都在上一次迭代中决策树残差减少的梯度方向上建立新的组合模型。

极端提升树(XGBoost)[18]改进了 GBDT,其使用牛顿法将目标函数展开到二阶优化,利用正则化项和特征采样技术降低模型复杂度并防止过拟合,累加 M 个决策树的预测结果作为最终的预测。同时,XGBoost 的 Block 结构能够实现特征排序,确定最佳分裂点的工作并行化,提高了模型的训练效率。

2.3.3 Stacking

Stacking[19]是一种分层的集成框架。首先在第一层使用训练集训练出若干个初级学习器,然后将初级学习器的学习结果作为次级学习器的输入数据集进行训练,得到最终的结果。为了防止过拟合,Stacking 通过 k 折交叉验证的方法,利用没有参与训练初级学习器的样本生成次级学习器的训练数据集。

3 S-FWS

本文提出的情感分析方法 S-FWS 的主要工作分为新词发现、混合特征提取和改进 Stacking 集成模型构建 3 部分。具体结构如图 1 所示。

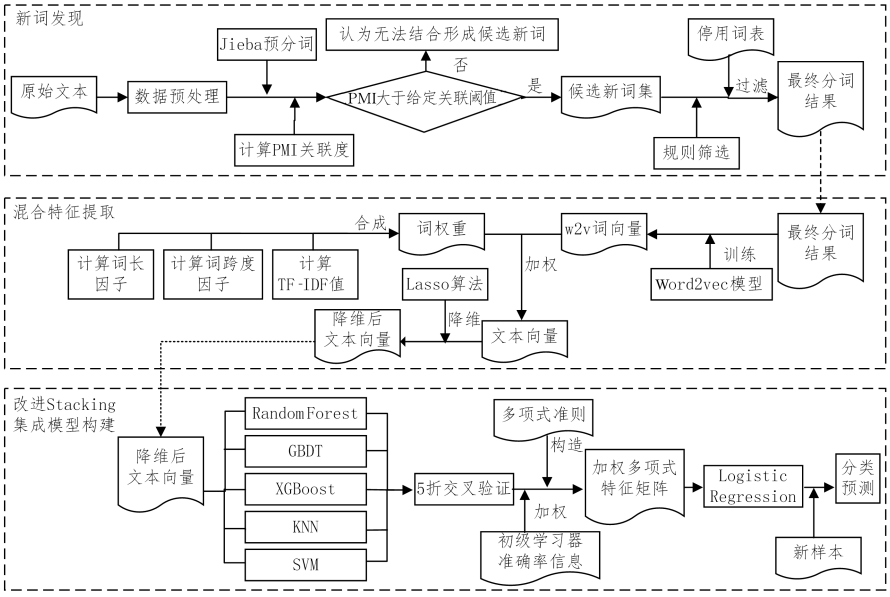

图 1 S-FWS 的结构

Fig. 1 Structure of S-FWS

3.1 新词发现

鉴于中文社交媒体的特殊性,在不同时段和不同舆情事件下往往会衍生出许多新潮的、不规范的词语,这些词语对句意和情感表达非常重要,但难以被正确捕捉并划分。针对此问题,S-FWS 首先使用 Jieba 库的精确模式对文本进行预分词,保证各词组被细粒度地划分开;然后使用 PMI 算法遍历计算每个词组与其邻接词组的关联程度,当某两个词组 d_i 和 d_j 的 $PMI(d_i,d_j)$ 大于设定阈值 $Thre$ 时,则将它们合并得到一个候选新词 $[d_i+d_j]$。值得注意的是,当某词组组合高频出现或阈值 $Thre$ 设定过低时,可能会导致很多词组被无效合并。如"今天/去/药店"被合并为"今天去药店",这个候选新词对文本分类并没有帮助作用,反而可能会损失时间或地点信息,因此需要对候选新词集做进一步的筛选。

遍历候选新词集,对于由多个词合并得到的候选新词 $[d_1,d_2,\cdots,d_n](n\geqslant 3)$,遵循以下拆分规则:

(1)若 $PMI(d_{i-1},d_i)=PMI(d_i,d_{i+1})$,保留原候选新词组合;

(2)若 $PMI(d_{i-1},d_i)<PMI(d_i,d_{i+1})$,按 d_i 左右进行拆分,保留 $[d_i,d_{i+1}]$,移除 $[d_{i-1},d_i]$;

(3)若 $PMI(d_{i-1},d_i)>PMI(d_i,d_{i+1})$,按 d_i 左右进行拆分,保留 $[d_{i-1},d_i]$,移除 $[d_i,d_{i+1}]$。

完成预分词和新词发现后,将百度、哈尔滨工业大学以及四川大学机器智能实验室停用词表合并导入 Python 中完成停用词过滤。

3.2 混合特征提取

每个词语的分类能力不同,给定文本中任一词语 d_i,用其词长信息 $WL(d_i)$、词跨度因子 $WS(d_i)$ 及其 TF-IDF 值衡量其重要度权重。其中,词长信息公式的表示如下:

$$WL(d_i)=length(d_i) \tag{7}$$

用 $Last(d_i)$ 和 $First(d_i)$ 表示 d_i 在文本中最后一次和首次出现的位置,用 All 表示文本中最后一个词的位置,词跨度因子公式表示如下:

$$WS(d_i)=\frac{Last(d_i)-First(d_i)}{All} \tag{8}$$

进而,词语 d_i 的重要度权重计算公式为:

$$MixWeight(d_i)=\log(a\times WL(d_i)+b\times WS(d_i)+c\times TFIDF(d_i)) \tag{9}$$

随后应用 Word2vec 的 Skip-gram 模型训练文本向量,将词重要度权重与其 Word2vec 词向量相乘得到加权词向量[20]。最后累加文本中每个词的加权词向量并求均值得到文本向量 TextVector,计算公式如下:

$$TextVector=\frac{\sum_i^n w2v(d_i)\times MixWeight(d_i)}{n} \tag{10}$$

通过多组预训练实验，最终确定参数 a 为 1.2，b 为 1.0，c 为 1.5，从而式(10)可表示为：

$$TextVector=\sum_{i}^{n}w2v(d_i)\times\log(1.2\times WL(d_i)+1.0\times WS(d_i)+1.5\times TFIDF(d_i))/n \tag{11}$$

得到初始文本向量后，为了避免过拟合和特征冗余，使用 Lasso 算法对其进行降维，Lasso 损失函数如下：

$$L(\beta)=\|y-X^{\mathrm{T}}\beta\|_2^2-\lambda\|\beta\|_1 \tag{12}$$

在 Spark 系统下，可以使用交替方向乘子法(Alternating Direction Method of Multipliers，ADMM)实现 Lasso 求解的分布式计算，目标增广拉格朗日函数为：

$$L(\beta,z,v)=\|y-X^{\mathrm{T}}\beta\|_2^2+\lambda\|z\|_1+\frac{\rho}{2}\|\beta-z+v\|_2^2 \tag{13}$$

迭代求解结果如下：

$$\begin{cases}\beta^{t+1}=(2X^{\mathrm{T}}X+\rho I_p)^{-1}(2X^{\mathrm{T}}y+\rho(z^t-v^t))\\ z^{t+1}=\mathrm{sign}(\beta^{t+1}+v^t)\left(\beta^{t+1}+v^t-\dfrac{\lambda}{\rho}\right)\\ v^{t+1}=v^t+\beta^{t+1}-z^{t+1}\end{cases} \tag{14}$$

3.3 改进 Stacking 集成模型构建

虽然 Stacking 的性能优越，但仍有改进之处。其中，初级学习器的预测结果是作为输入直接影响次级学习器的：初级学习器的正确预测显然能够帮助次级学习器进行决策；而错误预测则将错误信息传递给次级学习器，从而降低其泛化能力。如果次级学习器在训练时能够考虑初级学习器各自的学习情况，则会重点关注泛化性能强的初级学习器所对应的输入信息，将有效提高预测准确率。针对以上分析，本文改进了 Stacking：首先使用类概率特征替代类别特征，然后根据各初级学习器的准确率信息对类概率特征进行加权处理，进而构造多项式特征以增强特征表达，最后再将特征输入到次级学习器中完成训练。具体计算方式如下。

首先，计算各初级学习器的分类准确率：

$$Accuracy_i=\frac{\sum_{i=1}^{N}I(\hat{y}_i=y_i)}{N} \tag{15}$$

其中，$\hat{y}_i$ 为初级学习器对样本 i 的预测标签，y_i 则为真实标签。

其次，对 $Accuracy_i$ 进行标准化操作，从而满足概率密度分布，得到学习器 T_i 的权重 $Weight_i$：

$$Weight_i=\frac{Accuracy_i}{\sum_{k=1}^{m}Accuracy_k} \tag{16}$$

然后，对初级学习器 T_i 的类概率预测结果 $T_i(x_j)$ 进行加权处理，得到 $T_i(x_j)'$：

$$T_i(x_j)'=T_i(x_j)\times Weight_i \tag{17}$$

最后，基于加权后的类概率特征[$T_1(x_j)'$，$T_2(x_j)'$，…，$T_k(x_j)'$]构造多项式特征，使其特征表达能力得到增强。

对于初级学习器 T_i，若分类准确率 $Accuracy_i$ 越高，则权重 $Weight_i$ 越高，加权后的预测结果能够为次级学习器提供更多的正确信息，而学习情况较差的初级学习器的影响将被削弱。

遵循“好而不同”的集成原则，选择优秀且异质的随机森林 RF、梯度上升树 GBDT、XGBoost 作为初级学习器，降低模型的方差和偏差。同时，为了增加分类器多样性，在 Stacking 的初级学习器组中引入基于距离计算的 KNN 以及基于超平面划分的 SVM 模型。选定可解释性强的逻辑回归 LR 模型作为次级学习器，规避过拟合风险。改进 Stacking 算法的伪代码如算法 1 所示。

算法 1 改进 Stacking 算法的伪代码

输入：Train_data $D=\{x_i,y_i\}_{i=1}^{m}$

输出：Meta_classifier LR

1. Set Base_classifier

   $T_1=RF$，$T_2=GBDT$，$T_3=XGBoost$，$T_4=KNN$，$T_5=SVM$

2. Step1：Learn the Base_classifiers //第一层

3. For k=1 to 5 do

4. Learn T_k based on D

5. Calculate $Accuracy_k$ and $weight_k$ of T_k

6. Step2：Construct New Dataset of Predictions

7. For i=1 to m do //重要度加权

8. $D_T=\{x_i',y_i\}$，where $x_i'=\begin{Bmatrix}T_1(x_i)*weight_1\\ T_2(x_i)*weight_2\\ \cdots\cdots\\ T_5(x_i)*weight_5\end{Bmatrix}$

9. Construct Polynomial Features D_T' Based on D_T

10. Step3：Learn the Meta_classifier //第二层

11. Learn LR based on D_T'

12. Return LR

4 实验及结果分析

4.1 数据采集

本文通过 Python 自定义爬虫抓取新浪微博平台上“人民日报”账号在新冠肺炎疫情防控期间发布的疫情相关微博下其他用户发表的热门评论，共计 20 685 条，时间为 2020 年 1 月 23 日至 3 月 23 日共计两个月，并对这些评论进行人工情感标记，其中积极情绪评论有 10 183 条，消极情绪评论有 10 502 条(记为数据集 I)。同时，均衡采样 Datafountain 平台发布的疫情网民情绪识别博文 30 000 条(记为数据集 II)以及《战狼 II》上映相关舆情影评 60 000 条(记为数据集 III)进行验证。

4.2 实验环境

使用 Python 语言和 Spark 分布式系统进行实验，相关环境配置和软件版本如表 1 所列。

表 1 实验环境

Table 1 Experimental environment

名称	版本号或配置信息
Python	3.80
Java	1.8.0
Scala	2.12.10
Spark	3.0.1
Hadoop	3.3.0
操作系统	Windows10
集群节点数	1 个主节点，3 个从节点
节点配置	8GB 内存，四核处理器

4.3 模型评估指标

准确性能够很直观地反映一个分类模型的优劣，分类任务中常用的模型评价指标有正确率(Accuracy)、查准率(Precision，P)、召回率(Recall，R)等。本文综合考虑查准率 P 和召回率 R，使用 F1-score 来评价各分类模型的准确性能，计算式为：

$$P=\frac{TP}{TP+FP} \tag{18}$$

$$R=\frac{TP}{TP+FN} \tag{19}$$

$$F1\text{-}score=\frac{2\times P\times R}{P+R} \tag{20}$$

其中，TP 为将正类预测为正类的样本数，FN 为将负类预测为负类的样本数，FP 为将负类预测为正类的样本数。

由于时间资源有限，算法的运行速度也十分重要。本文在单机模式和分布式下的实验均以秒为单位来记录算法的运行耗时。

4.4 准确性能实验

针对实验数据集 I、数据集 II、数据集 III，应用 S-FWS 方法完成特征提取和模型建立，并与 TF-IDF 和 Word2vec 等 4 种特征提取方法进行对比。图 2 给出了 5 种特征提取方法在 3 个数据集下的 F1-score。由图 2 可以看出，S-FWS 在 3 个数据集下的 F1-score 分别达到了 0.893，0.807 和 0.865，均为最高值。这表明混合特征提取模式挖掘出了更多的文本信息，更加适用于中文社交媒体领域。

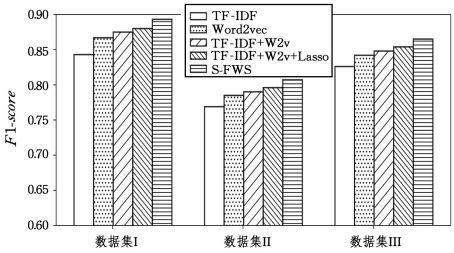

图 2 各特征提取法的 F1-score

Fig. 2 F1-score of each feature extraction algorithm

进而将 S-FWS 方法构建的改进 Stacking 模型与 Stacking，StackingC[21]，Torikca[22]，Stacking-PMLR[23]，Stacking-MLLR[24]等先进集成模型进行对比。图 3—图 5 给出了各模型在 3 个数据集上的 F1-score。由图可知，StackingC 模型的准确性能最差，而 S-FWS 在 3 个数据集的 F1-score 分别达到了 0.893，0.807 和 0.865，均高于其他模型，准确性能最佳。其中，相比传统 Stacking 模型，本文提出的 S-FWS 模型的 F1-score 平均提高了 0.0221。

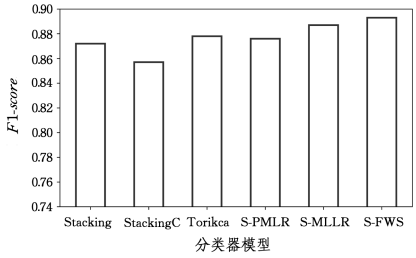

图 3 数据集 I 下各模型的 F1-score

Fig. 3 F1-score of each model on dataset I

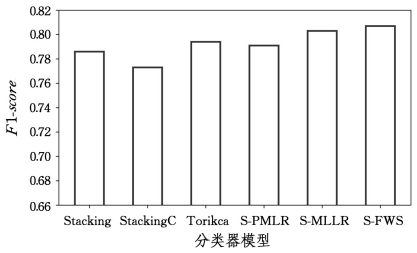

图 4 数据集 II 下各模型的 F1-score

Fig. 4 F1-score of each model on dataset II

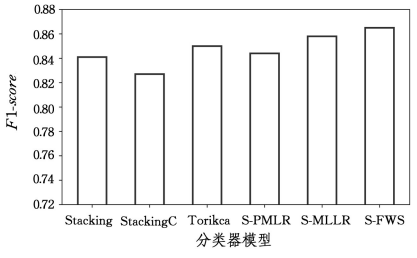

图 5 数据集 III 下各模型的 F1-score

Fig. 5 F1-score of each model on dataset III

4.5 分布式加速实验

分别在单机模式和 Spark 下完成以上集成模型的训练和预测，研究分布式系统加速算法的优越性，运行耗时情况如表 2—表 4 所列。

表 2 数据集 I 下各分类器的运行耗时情况

Table 2 Running time of each classifier on dataset I

(单位：s)

分类器	单机模式		Spark 单节点		Spark 两节点	
	数据处理	算法分类	数据处理	算法分类	数据处理	算法分类
Stacking	173.97	58.59	226.18	79.20	118.50	36.43
StackingC	93.32	32.61	121.34	44.12	63.66	20.32
Torikca	307.21	100.03	399.47	135.14	209.10	62.12
S-PMLR	132.78	45.57	172.71	61.62	90.49	28.35
S-MLLR	209.03	68.12	271.84	91.86	142.34	42.33
S-FWS	181.32	63.83	235.82	86.27	123.50	39.67

表 3 数据集 II 下各分类器的运行耗时情况

Table 3 Running time of each classifier on dataset II

(单位：s)

分类器	单机模式		Spark 单节点		Spark 两节点	
	数据处理	算法分类	数据处理	算法分类	数据处理	算法分类
Stacking	263.93	88.76	343.31	119.93	179.67	55.13
StackingC	142.75	49.17	185.78	66.48	97.27	30.59
Torikca	465.11	152.82	604.84	206.41	316.47	94.85
S-PMLR	201.97	69.40	262.76	93.79	137.54	43.13
S-MLLR	317.00	104.20	412.30	140.77	215.76	64.70
S-FWS	274.45	96.42	356.89	130.27	186.83	59.98

表 4 数据集 III 下各分类器的运行耗时情况

Table 4 Running time of each classifier on dataset III

(单位：s)

分类器	单机模式		Spark 单节点		Spark 两节点	
	数据处理	算法分类	数据处理	算法分类	数据处理	算法分类
Stacking	529.41	179.93	688.43	243.11	360.23	111.66
StackingC	288.23	101.35	374.90	136.92	196.20	62.94
Torikca	935.53	306.67	1216.39	414.14	636.36	190.24
S-PMLR	408.03	141.62	530.64	191.29	277.69	87.90
S-MLLR	637.16	211.13	828.51	285.13	433.47	130.97
S-FWS	562.52	194.49	731.48	262.66	382.71	120.68

观察表 2－表 4，通过纵向对比可以发现：模型的运行耗时随集成复杂度的提高而增长，其中 Torikca 模型层数最多，相应耗时也最长。由于 S-FWS 需要对次级学习器的输入特征进行加权和多项式构造，因此相比传统 Stacking 耗时有所增长，但平均仅增长 5.762%，增幅并不明显，仍具备良好的耗时性能。

通过横向对比可以发现，因为 Spark 本身的资源调度需要时间，所以当 Spark 集群只有一个工作节点时，其运行效率甚至低于单机模式；当 Spark 集群增加为两个工作节点后，算法的数据处理和预测分类速度都得到了提高，此时总运行效率平均提升了 33.37%。

继续增加 Spark 集群工作节点，查看 S-FWS 在不同工作节点数下的耗时情况，如图 6 所示。由图 6 可以看出，S-FWS 运行耗时随集群工作节点数的增加而减少，但曲线斜率呈减小趋势，即运行耗时减幅逐渐放缓。当有 4 个工作节点时，总运行效率平均提升能够达到 63.62%。

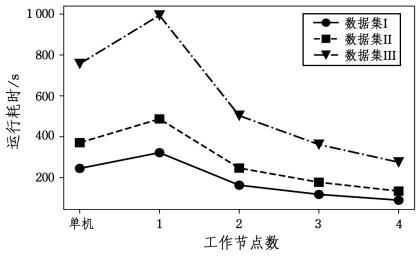

图 6 不同节点数下运行耗时情况

Fig. 6 Running time of different nodes

综上，S-FWS 仍保持了较短的耗时，且 Spark 分布式系统能够显著提高算法的运行速度。

结束语 本文基于 Spark 分布式平台，使用 S-FWS 方法实现了对舆情事件下中文社交媒体文本的情感大数据分析。从实验结果来看，应用 S-FWS 方法提取的文本特征信息更充分，构建的改进 Stacking 集成模型的分类能力更强，适用于中文社交媒体文本分类。并且 Spark 分布式系统能够大幅提高算法的运行效率，同时在模型的调参工作中发挥着巨大的提速作用。但本文方法仍有改进空间，如集成更加准确和高效的模型等，在今后的工作中将进行更深入的研究。

参 考 文 献

[1] PANG B, LEE L. Opinion mining and sentiment analysis[J]. Foundations & Trends in Information Retrieval, 2008, 2(1/2): 1-135.

[2] PANG B, LEE L, VAITHYANATHAN S. Thumbs up? Sentiment classification using machine learning techniques[C]// Proceedings of 2002 Empirical Methods in Natural Language Processing. 2002: 79-86.

[3] WANG T, LI M. Research on Comment Text Mining Based on LDA Model and Semantic Network[J]. Journal of Chongqing Technology and Business University(Natural Science Edition), 2019, 36(4): 9-16.

[4] YUN W, GAO Q. An Ensemble Sentiment Classification System of Twitter Data for Airline Services Analysis[C]// IEEE International Conference on Data Mining Workshop. IEEE, 2015: 1318-1325.

[5] ALREHILI A, ALBALAWI K. Sentiment analysis of customer reviews using ensemble method[C]// Proc of International Conference on Computer and Information Sciences. Piscataway, NJ: IEEE press, 2019: 1-6.

[6] ZHANG Y, ZHOU Y, LU H, et al. Traffic Network Flow Prediction Using Parallel Training for Deep Convolutional Neural Networks on Spark Cloud[J]. IEEE Transactions on Industrial Informatics, 2020(99): 1-1.

[7] ELZAYADY H, BADRAN K M, SALAMA G I. Sentiment Analysis on Twitter Data using Apache Spark Framework[C]// 2018 13th International Conference on Computer Engineering and Systems (ICCES). 2018: 171-176.

[8] YANG L Y, WANG Y Z. Application of Spark in Sentiment Analysis of Ensemble Learning Text[J]. Computer Applications and Software, 2020, 37(6): 130-134.

[9] WANG W H, JIN L J. Sentiment Analysis Ensemble Algorithm Based on Spark[J]. Journal of Zhejiang University of Technology, 2020, 48(4): 405-410, 434.

[10] GRBIC D, HAFFERTY F W, HAFFERTY P K. Medical School Mission Statements as Reflections of Institutional Identity and Educational Purpose: A Network Text Analysis[J]. Academic Medicine: Journal of the Association of American Medical Colleges, 2013, 88(6): 852-860.

[11] ZHU A Q, LI S, TANG X D. Parallel FP_growth Association Rules Mining Method on Spark Platform[J]. Computer Science, 2020, 47(12): 139-143.

[12] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing[C]// Usenix Conference on Networked Systems Design & Implementation. 2012: 15-28.

[13] PECINA P, SCHLESINGER P. Combining association measures for collocation extraction [C]// Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. Stroudsburg: ACL, 2006: 651-658.

[14] HE J F, ZHAO H, HE X M. Suspicious Person Text Representation Method Based on Improved TF-IDF[J]. Computer Engineering and Design, 2021, 42(2): 396-401.

[15] ZHANG D W, XU H, SU Z C, et al. Chinese comments sentiment classification based on word2vec and SVMperf[J]. Expert Systems with Applications, 2015, 42(4): 1857-1863.

[16] BREIMAN L. Random Forests [J]. Machine Learning, 2001, 45(1): 5-32.

[17] FRIEDMAN J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5): 1189-1232.

[18] CHEN T, GUESTRIN C. XGBoost: a scalable tree boosting system[J]. International Conference on Knowledge Discovery and Data Mining, 2016, 1(1): 785-794.

[19] DŽEROSKI S, ŽENKO B. Is combining classifiers with stacking

better than selecting the best one? [J]. Machine Learning, 2004, 54(3): 255-273.

[20] CHU T Z, CHENG L, WONG H S. Corpus-based topic diffusion for short text clustering[J]. Neurocomputing, 2018, 275: 2444-2458.

[21] ASHRAF M, ZAMAN M, AHMED M. Using Ensemble StackingC Method and Base Classifiers to Ameliorate Prediction Accuracy of Pedagogical Data [J]. Procedia Computer Science, 2018, 132: 1021-1040.

[22] MENAHEM E, ROKACH L, ELOVICI Y. Troika—An improved stacking schema for classification tasks[J]. Information Sciences, 2009, 179(24): 4097-4122.

[23] DAI H, WU W K, LI J C, et al. Incorporating Feature Selection in the Improved Stacking Algorithm for Online Learning Analysis and Prediction[J]. Engineering Letters, 2020, 28(4): 1011.

[24] JIANG M, LIU J, ZHANG L, et al. An improved stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms[J]. Physica A: Statistical Mechanics and its Applications, 2020, 541: 122272.

DAI Hong-liang, born in 1978, Ph. D, professor, postdoctoral supervisor, is a member of China Computer Federation. His main research interests include machine learning and big data analysis.

DAI Hong-ming, born in 1978, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include machine learning and big data analysis, and software engineering.