Central University of Finance and Economics

中央财经大学

大数据分析分布式计算

基于 Spark 的文本分类

吴宇翀

2021210793

email@wuyuchong.com

指导老师：李丰

2022 年 1 月 15 日

1 摘要

我们使用主题模型进行对电影评论进行文本挖掘，之后进行情感分类模型的训练。在文本预处理阶段，我们尝试使用词编码和词向量的方式，在训练阶段，我们构建了 DNN、LSTM、BERT 等多个深度学习模型进行训练，并进行了模型比较，最终达到了 90% 的准确率。最后，为了进一步实现在超大文本集上进行训练，我们使用基于 Spark 的分布式算法在集群服务器上进行训练测试。[1]

模型	计算配置	用时	准确率	可拓展性
tokenize + DNN	阿里云服务器 Xeon 8 核 CPU 32G 内存	10 分钟	60%	低-单机
Word2Vec + LSTM	阿里云服务器 Xeon 8 核 CPU 32G 内存	2 小时	80%	低-单机
bert - 小型	阿里云服务器 Xeon 8 核 CPU 32G 内存	1 小时	86%	低-单机
bert - AL	阿里云服务器 Xeon 8 核 CPU 32G 内存	1.5 小时	88%	低-单机
bert - 标准	阿里云服务器 Xeon 8 核 CPU 32G 内存	3 小时	90%	低-单机
spark - logit	中央财经大学大数据高性能分布式集群	10 分钟	83%	高-集群
spark - 决策树	中央财经大学大数据高性能分布式集群	40 分钟	85%	高-集群
spark - 梯度助推树	中央财经大学大数据高性能分布式集群	20 分钟	87%	高-集群
spark - 随机森林	中央财经大学大数据高性能分布式集群	1.5 小时	89%	高-集群

[1] 分布式模型在该小型数据集上没有优势，进行此项的意义在于对大型文本数据集可拓展性的技术储备，仅有在文本量级超过单机可承载上限时，分布式计算才具备意义

目录

2 数据集介绍

我们选择了 IMDB 的电影评论文本数据进行大数据建模研究。

IMDB 是一个隶属于亚马逊公司旗下的世界著名互联网电影资料库（Internet Movie Database）。它有着关于电影演员、电影、电视节目、电视明星和电影制作的在线数据，包括了影片的众多信息、演员、片长、内容介绍、分级、评论等，在电影评论评分时被广泛使用。IMDB 的论坛也十分活跃，除每个数据库条目都有留言板之外，还有关于多种多样的主题的各种综合讨论版。

我们将 IMDB 的电影评论文本用于自然语言处理的二元情感分类。我们使用 5 万条标有积极和消极标签的真实用户电影评论文本构建情感分类模型。即使用深度学习算法预测评论为正面或是负面。

我们使用的文本为多语言文本，其中英文文本数量占绝大多数比例。

评论	情感
One of the other …	positive
A wonderful littl…	positive
I thought this wa…	positive
Basically there's…	negative
Petter Mattei's "…	positive
Probably my all-t…	positive
I sure would like…	positive

3 分布式训练

我们使用 `pyspark` 进行分布式训练。分布式不同于单机训练，而是通过集群上许多的计算机节点同时进行训练。对于文本量很大的数据集而言，单机可能不具备足够的内存和 CPU 资源进行训练，借助于分布式系统，我们能调度集群计算资源进行计算。`pyspark` 是 `spark` 在 `python` 下的实现，它使用 Zookeeper、hadoop 作为底层，通过 MapReduce 的方式将大的计算任务拆解成为一个个小的任务，分发到每个计算机节点上进行计算。

3.1 环境启动

- 通过 YARN 资源调度系统提交到作业队列：`spark-submit --master yarn`
- 由于在 UDF（用户自定义）函数中使用了第三方包，需要将其发送至集群中的每个计算节点 `--py-files gensim.zip`
- 队列计算完成后将结果重定向输出 `> output.txt`

3.2 数据读取

由于数据为逗号分隔的 csv 格式，在文本列出现混淆。我们使用 pandas 进行读取后再转换为 spark DataFrame 格式

3.3 文本清洗

1. 去除标签
   - 将一些网页 HTML 特有的标签进行去除，如 `p` `br` 等
2. 去除标点符号
   - 将常用标点符号进行去除，如 ！ ; 等
3. 去除多余的空格
   - 删除无意义的连续性空格
4. 去除数字
   - 由于数字对文本情感识别作用小，我们选择将其删去
5. 去除停用词
   - 对意义较小的常用词进行删除
6. 去除过短的词汇
   - 由于英文中过短的字符一般意义较小，我们选择将其删去
7. 大小写统一
   - 大小写代表同一词汇，需要进行统一

review	sentiment	clean_text	label
” Så som i himmel…	positive	som himmelen spec…	1.0
”A Thief in the N…	positive	thief night film …	1.0
”A bored televisi…	negative	bore televis dire…	0.0
”A death at a col…	negative	death colleg camp…	0.0
”A wrong-doer is …	positive	wrong doer man le…	1.0

3.4 文本特征工程

词频-逆文档频率（TF-IDF）是一种广泛用于文本挖掘的特征向量化方法，它反映了单个词汇相对于语料库中文档的重要性。我们用表示 t 代表词汇，用 d 代表表示文档，用 D 表示语料库。词频 $TF(t,d)$ 是该词在文档 d 中出现的次数，而文档频率 $DF(t,D)$ 是包含该词的文档的数量。如果我们只使用词频来衡量

重要性，很容易过分强调那些出现频率很高但几乎没有关于文档的信息的词，例如 “这”“的” 等词汇。如果一个术语在语料库中经常出现，则意味着它不包含有关特定文档的特殊信息。逆文档频率是一个术语提供多少信息的数值度量：

$$IDF(t, D) = \log \frac{|D| + 1}{DF(t, D) + 1}$$

其中 $|D|$ 是语料库中的文档总数。

由于使用对数，如果一个词出现在所有文档中，它的 IDF 值变为 0，因此使用平滑词以避免对语料库之外的词除以零。TF-IDF 是 TF 和 IDF 的乘积：

$$TFIDF(t, d, D) = TF(t, d) \cdot IDF(t, D)$$

在 TF 的基础上，我们使用改进版的 HashingTF 进行处理。HashingTF 将词汇转换为固定长度的特征向量。HashingTF 利用散列表应用哈希函数映射词汇到索引，之后通过映射的函数计算词频，能有效降低 TF 在大型语料库所需的时间。

我们从一组句子开始，将每个句子分成单词，构建词袋，使用 HashingTF 将句子散列成特征向量，使用 IDF 重新缩放特征向量，然后将我们的特征向量传递给学习算法。

clean_text	filtered_tokens	vectorizedFeatures
som himmelen spec…	[som, himmelen, s…	(61505,[1,5,7,8,1…
thief night film …	[thief, night, fi…	(61505,[0,1,7,20,…
bore televis dire…	[bore, televis, d…	(61505,[0,4,7,14,…
death colleg camp…	[death, colleg, c…	(61505,[0,6,8,15,…
wrong doer man le…	[wrong, doer, man…	(61505,[1,2,4,6,9…

3.5 训练模型

我们首先使用简单的 logistic 模型进行拟合，在训练集上进行拟合，之后在测试集上验证模型的效果。

真实值	预测值
0.0	0.0
0.0	0.0
1.0	1.0
1.0	1.0
0.0	0.0
1.0	1.0
0.0	1.0

真实值	预测值
0.0	0.0

3.6 模型预测

我们准备了两个测试用例来验证模型是否有效。

1. 我喜欢这部电影
2. 它很差劲

模型对前一个句子的分类结果为积极，对一个句子的分类结果为消极。

3.7 模型调参与比较

在 logistic 模型的基础上，我们还搭建了随机森林模型、梯度助推树模型、决策树模型。

我们使用网格搜索的方式对几个模型的超参数进行调整，选取最优的模型

4 Word2Vec

Word2Vec 在自然语言处理中作为基础的一步，在它的基础上文本分类、相似查询、多语言翻译、问答系统等应用获得了很好的效果。

4.1 原始算法

Word2Vec 将单词表示为低维度向量，在训练得当的情况下，意思相近的单词在具有相似的向量表示。且 Word2Vec 这种向量表示具有良好的加减性质，如国王 - 男人 + 女人能够得到近似王后的向量。

词汇的向量化表示一般使用一个较浅的神经网络进行训练，训练的目标即为尽可能地使得每个词汇的向量表示能够用来预测上下文词汇，或是被周围的词汇预测，基于此一般有两种训练 Word2Vec 的方法：CBOW 和 Skip-gram。CBOW 使用上下文词汇的向量表示作为输入训练目标词汇，Skip-gram 训练目标词汇的向量表示以达到最好的预测上下文词汇的效果，两种方法并没有本质上的区别。

此处应有插图

以 Skip-gram 为例，给定一个按顺序的词汇序列 $w_1, w_2, w_3, \dots, w_T$，训练的目标即为最大化对数似然函数：

$$J(\Omega) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p\left(w_{t+j} \mid w_t\right)$$

其中 c 是上下文滑动窗口的尺寸，$p\left(w_{t+j} \mid w_t\right)$ 是给定 w_{t+j} 时中心词出现的概率。

在原始的惯用的算法中，使用一个简单的只含一个隐藏层的神经网络进行训练，使用 softmax 作为输出层的激活函数：

$$p\left(w_O \mid w_I\right) = \frac{\exp\left(\left\langle \mathbf{v}_{in}^{w_I}, \mathbf{v}_{out}^{w_O}\right\rangle\right)}{\sum_{w=1}^{V}\exp\left(\left\langle \mathbf{v}_{in}^{w_I}, \mathbf{v}_{out}^{w}\right\rangle\right)}$$

其中，V 表示代表语料库的大小，$\langle\cdot,\cdot\rangle$ 代表两个向量的内积。

这种计算方式与语料库的大小成正比，但语料库非常大的时候是比较耗时的。

4.2 改进算法

通过负采样的方式，我们可以近似对数的 softmax 函数：

$$\begin{aligned}\log p\left(w_O \mid w_I\right) \approx{}& \log\sigma\left(\left\langle \mathbf{v}_{in}^{w_I}, \mathbf{v}_{out}^{w_O}\right\rangle\right)\\ &+\sum_{k=1}^{K}\mathbb{E}_{w_k\sim P_n(w)}\left[\log\sigma\left(-\left\langle \mathbf{v}_{in}^{w_I}, \mathbf{v}_{out}^{w_k}\right\rangle\right)\right],\end{aligned}$$

4.3 分布式实现

将 Word2Vec 进行分布式实现，在集群中的各个计算节点之间分配计算任务。

上至 32 节点