汽车行业用户观点主题及情感识别

——数据科学导论期末报告

何秦兴 PB16110299

吴雨菲 PB15020525

张劲暾 PB16111485

2018 年 12 月 7 日

目录

1 题目描述

1.1 题目背景

随着政府对新能源汽车的大力扶植以及智能联网汽车兴起都预示着未来几年汽车行业的多元化发展及转变。汽车厂商需要了解自身产品是否能够满足消费者的需求，但传统的调研手段因为样本量小、效率低等缺陷已经无法满足当前快速发展的市场环境。因此，汽车厂商需要一种快速、准确的方式来了解消费者需求。

1.2 题目任务

本赛题提供一部分网络中公开的用户对汽车的相关内容文本数据作为训练集，训练集数据已由人工进行分类并进行标记，参赛队伍需要对文本内容中的讨论主题和情感信息来分析评论用户对所讨论主题的偏好。讨论主题可以从文本中匹配，也可能需要根据上下文提炼。

1.3 数据说明

训练集数据中主题被分为 10 类，包括：动力、价格、内饰、配置、安全性、外观、操控、油耗、空间、舒适性。

情感分为 3 类，分别用数字 0、1、-1 表示中立、正向、负向。

content_id 与 content 一一对应，但同一条 content 中可能会包含多个主题，此时出现多条记录标注不同的主题及情感，因此在整个训练集中 content_id 存在重复值。

字段名称	类型	描述	说明
content_id	Int	数据 ID	/
content	String	文本内容	/
subject	String	主题	提取或依据上下文归纳出来的主题
sentiment_value	Int	情感分析	分析出的情感
sentiment_word	String	情感词	情感词

1.4 评测标准

本赛题采用 F1-Score 评价方式。按照“主题 + 情感分析”识别数量和结果（是否正确）来进行判断，参赛者需要识别文本中可能包含的多个“主题”。匹配识别结果: Tp: 判断正确的数量；Fp: 判断错误或多判的数量；Fn；漏判的数量.

当提交的一条数据结果包含“主题 + 情感值”，如果参赛者对“主题 + 情感”的判断结果完全正确则计入 Tp，如果对“主题”或“情感值”的判断结果错误则计入 Fp；如果参赛者未能对某一数据文本判断“主题”或“情感值”给出判断结果，则此条数据不能包含在结果文件中；如果参赛者识别出的“主题 + 情感值”数量少于测试样本中实际包含的数量，或未对某个测试样本数据给出结果，缺少的数量计入 Fn；如果参赛者识别出的“主题 + 情感值”数量多于测试样本中实际包含的数量，超出的数量计入 Fp

$$\text{准确率：} P = \frac{T_p}{T_p + F_p}$$

$$\text{召回率：} R = \frac{T_p}{T_p + F_n}$$

$$\textbf{F1-Score：} F1 = \frac{2PR}{P + R}$$

2 题目分析与方案设计

2.1 对数据的基本特征分析和设计思路

数据基本特征和任务分析：短文本情感分类与主题挖掘，通过将文本映射到向量空间，并将需要预测的情感和主题用独热码表示或编号为类别，将问题转化为学习从文本特征表达到类标签的的映射关系。

文本的数量特征对模型设计的影响：文本长度普遍较短，可以提取的特征较少，所以需要提取有效的特征；多标签问题：每个实例可能对应多个主题标签，而且对于不同的主题标签还有可能是不同的情感分类，这样多主题的数据大概占到总数据的 15%，这一点对于我们采用的主题、情感分开学习，不区分同一 ID 的不同数据实例的设计方案是不太友好的，参考成功的比赛选手的方案，应该将情感和主题组合得到 30 个标签进行学习，以解决多标签问题。数据数量偏少，只有不到一万条；数据质量差，很多主题情感标签模糊，对于很多正面或者负面的评论笼统标注为中性，中性情感标签明显多余另外两类，造成一定的学习过拟合问题，特别是对于 LSTM 模型的学习影响较大。

2.2 特征提取与模型选择

对与文本的特征映射，我们采取了两个思路方向，第一是采取将文本分词之后映射到一维的特征向量，比如使用 LDA 模型或者 TFIDF 模型，得到特征向量之后交给分类器学习；第二是先训练一个词向量模型，然后将文本映射为一个以词向量为行的特征矩阵，然后卷积池化得到特征向量，交给 LSTM 模型或者 CNN 模型学习。

对于从文本特征空间到标签的映射关系学习，我们将对于情感标签的学习和尝试了逻辑回归模型（LR）、朴素贝叶斯模型（NB）、多层感知机模型（MLP）和多层感知机

模型（MLP），最终选择将这些模型进行集成，用最终得到的集成模型分别对情感标签和主题标签进行学习分类。

3 方法说明

3.1 分词

我们首先去除了非中文符号，并且找到了一份通用的汉语停用词表，人工去除了其中可能与主题和情感取向有关的词语，然后分别调用了 jieba 分词包和 THULAC 分词包对评论进行分词预处理，jieba 分词的结果较为零散，而 THULAC 的分词结果更加接近于汉语习惯和常用词汇，但在同样的模型测试后，jieba 分词得到的效果略好于 THULAC，所以我们最终采用了 jiaba 包做分词处理。

3.2 特征提取

3.2.1 隐含狄利克雷主题模型 (LDA)

3.2.2 词袋模型（BOW）

3.2.3 TFIDF

3.2.4 输入 LSTM 的词向量矩阵

词向量是 NLP 中最基本的概念之一，词向量将抽象的语言符号化数学化。主要分为两种:

one-hot representation:

举个例子:

“话筒”表示为 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ⋯]

“麦克”表示为 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 ⋯]

每个词都是茫茫 0 海中的一个 1。

distributed representation:

形如 [0.792, −0.177, −0.107, 0.109, −0.542, ⋯]

Distributed representation 最大的贡献就是让相关或者相似的词，在距离上更接近了。向量的距离可以用最传统的欧氏距离来衡量，也可以用 cos 夹角来衡量。用这种方式表示的向量，“麦克”和“话筒”的距离会远远小于“麦克”和“天气”。可能理想情况下“麦克”和“话筒”的表示应该是完全一样的，但是由于有些人会把英文名“迈克”也写成“麦克”，导致“麦克”一词带上了一些人名的语义，因此不会和“话筒”完全一致。

我使用的就是分布式表示，即词嵌入（word embedding），我在实现时直接利用了 python gensim 包里的 word2vec。

Word2vec 是一组用于生成单词嵌入的相关模型。这些模型是浅层的双层神经网络，经过训练可以重建语言的语言环境。Word2vec 将大量文本作为其输入，并产生通常为几百维的向量空间，语料库中的每个唯一单词在空间中被分配相应的向量。单词向量位于向量空间中，使得在语料库中共享共同上下文的单词在空间中彼此非常接近地定位。

这次实验中我们是在数据集上直接训练的 Word2vec 模型，并没有借助外部数据训练，简单测试表现训练得到的模型对于主题相关词和情感相关词的判断直观上符合逻辑。然后我们将词向量维度设置为 100 维并测得每条评论的关键词长度最多为 80, 于是我们将每条评论中关键字的特征向量依次排列，不足的 padding 为零，得到一个 100*80 的矩阵作为 LSTM 模型的输入（带卷积池化层转换为一维特征向量）。

3.3 学习方法

3.3.1 逻辑回归模型（LR）

3.3.2 朴素贝叶斯模型（NB）

3.3.3 多层感知机模型（MLP）

3.3.4 随机森林模型（RF）

对于样本较少的且不太均衡的数据来说，是很容易发生过拟合的。此处使用随机森林的出发点在于对于不平衡的分类资料集来说，随机森林的方法可以平衡误差，且由于随机森林的 ensemble 特点，它可以产生高准确度的分类器。

随机森林即由很多决策树构成的森林，每棵决策树都是一个分类器（假设现在针对的是分类问题）。

随机森林的随机主要体现在两个方面：

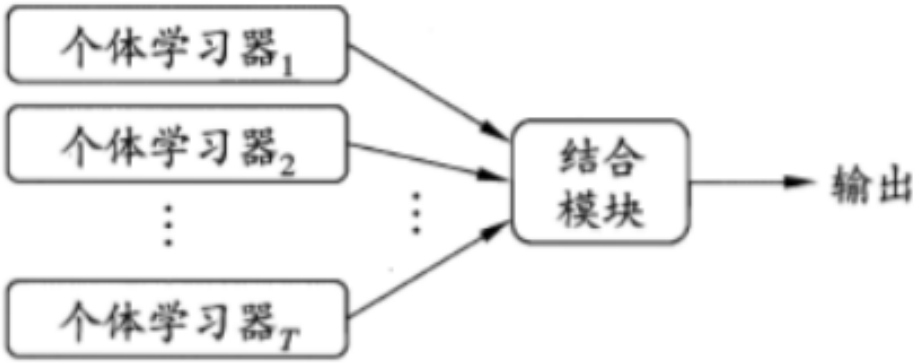
1. 数据选取的随机，类似于 bagging 算法中的自助采样法（bootstrap sampling），每一颗决策树都从 m 个数据中随机有放回地取 m 个数据，约有近三分之一样本的测试集，三分之二样本的训练集。

2. 属性的随机，传统的决策树从当前节点的所有属性中选取一个最优属性，而随机森林的决策树先取一个含 k 个属性的子集，再在里面取最优，属性的扰动增加了个体学习器的差异度，增强了模型的泛化性能。（西瓜书 P179,P180）

随机森林算法主要代码 (待补充)

3.3.5 集成模型（Ensemble）

集成学习通过构建并结合多个学习器来完成学习任务, 一般的结构为先产生一组个体学习器，再用某种策略把他们结合起来，典型的有 AdaBoost **算法、**bagging **算法和随机森林算法。现在也有人将当下主流的各种** NN models **进行集成来达到更强的泛化能力与强健性，这与集成学习的概念有所出入但是效果类似。(参考西瓜书** P171)



集成学习示意图

Boosting Boosting **是一族可将弱学习器提升为强学习器的算法。这一族算法的工作机制都是类似的：先从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的样本分布来训练下一个基学习器**

Bagging Bagging **算法以自助采样法（**bootstrap sampling**）为基础，从** m **个数据中随机有放回地取** m **个数据，约有近三分之一（** $lim_{m\to+\infty}\ (1-\frac{1}{m})^m$ **）的样本不会被选中，将这些样本作为测试集, 其余作为训练集。于是，我们可以采样出** T **个含** m **个训练样本的采样集，然后基于每个采样集训练出一个基学习器，再集成，这就是** Bagging **的基本流程。(参考西瓜书** P173，P178)

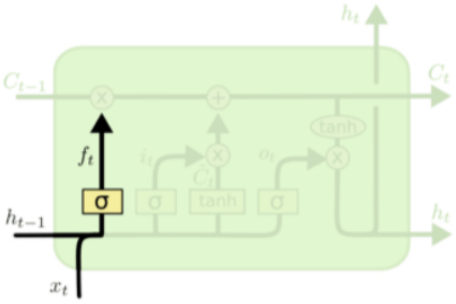
自助法在数据集较小时很有用，并且能从初始数据集中产生多个不同的训练集，这对集成学习有很大的好处。（西瓜书）在这次实验中我们使用 sklearn.ensemble **框架中的** VotingClassifier **集成模型集成了逻辑回归模型** (LR) **朴素贝叶斯模型** (NB) **多层感知机模型** (MLP) **随机森林模型** (RF) **四种模型对情感和主题进行分类学习。**

3.3.6 长短期循环神经网络（LSTM）

LSTM 网络非常适合基于时间序列数据进行分类，处理和预测，因为在时间序列中的重要事件之间可能存在未知持续时间的滞后。LSTM 能够捕捉到这些滞后的关联。

LSTM 单元有几种架构。通用架构由存储器单元，输入门，输出门和遗忘门组成。

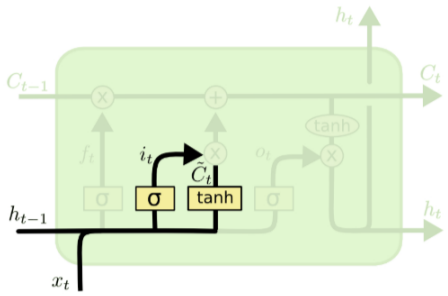
一些思考　有论文对主流的深度学习模型进行了比较，指出 LSTM 在各类任务中表现优异，有十足的健壮性（robust），唯独在关键词识别（keyphrase recognition）例如情感识别中表现不如其他（当然也不差）。我觉得原因在于句子的情感往往是鲜明的，反应在学习器的输出上的话这些输出值的分布应当不是很均匀的（趋向两级），LSTM 捕捉的前后关联自然是没有精准定位情感词来得简单有效。（加上本身中立数据较多，使过拟合更为明显）



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

遗忘门

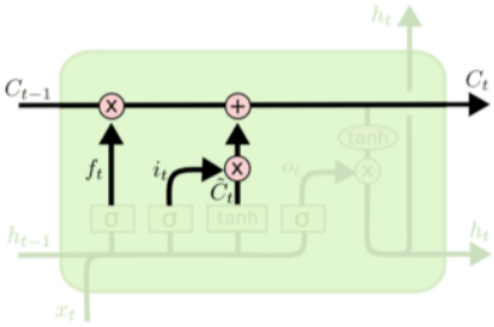
遗忘门取前一次的细胞状态 C_{t-1} 为输入，根据需要调权重输出 C_{t-1} 的一部分



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

输入门

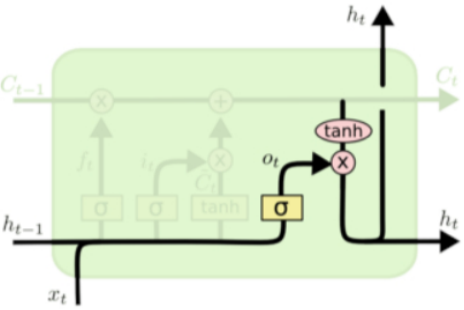
输入门决定让多少新的信息加入到细胞的状态中来



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

更新后的细胞状态 C_t

输入门加上遗忘门就是新的细胞状态 C_t



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

输出门

最后通过输出门，只输出我们想输出的部分，用于调节整个神经网络信息传递

```
        model = Sequential()
    model.add(Embedding(output_dim=vocab_dim,
                        input_dim=n_symbols,
                        mask_zero=True,
                        weights=[embedding_weights],
                        input_length=input_length))
    model.add(LSTM(output_dim=50, activation='tanh'))
    model.add(Dropout(0.8))
    model.add(Dense(3, activation='softmax'))
    model.add(Activation('softmax'))

    print('Compiling...')
    model.compile(      loss='categorical_crossentropy',
                                optimizer='adam',
                                metrics=['accuracy'],
                                sample_weight_mode='temporal')

    print("Train...")
    model.fit(  x_train, y_train,
                        batch_size=batch_size,
                epochs=n_epoch, verbose=1)

    print("Evaluate...")
    score = model.evaluate(x_test, y_test,batch_size=batch_size)
```

3.4 提升和改进的思路

首先是对于数据的利用，将给出的情感词利用起来拼在分词结果后面，加强对于情感词的学习，抵消分词不恰当的不利因素，在最终的结果中可以提升一到两个百分点。另外一些没来得及实验的提升设想： 1. **对得到的** TFIDF **向量做** PCA **主成分分析提升可区分性；** 2. **用引入** attention **机制的** LSTM **对情感分类进行学习；** 3. **在词这一层面上对相同情感标签的评论采样生成新的样例，改善类不平衡问题；** 4. **对于主题分类设置阈值，不仅仅是输出最有可能的主题标签，而是将可能性超过阈值的标签分条输出；** 5. **不要将主题和情感分开学习，而是组合得到** 30 **种标签去学习，因为数据本来的特征就**

是对于不同的主题可能有不同的情感态度，这样更接近问题的本质。

4 试验结果

4.1 线上比赛结果

最终最佳结果：A **榜**：0.60918770000；B **榜**：0.61174583000
线上其他选手最高成绩：A **榜**：0.66085330；B **榜**：0.66480136

4.2 各种方法的结果与分析

BOW + MLP	0.46138483000
LDA + LR	0.44940080000
LDA + MLP	0.50000000000
LDA + MLP + LSTM（情感）	0.41810918000
LDA + RF	0.53928095000
LDA + RF + LSTM（情感）	0.44207722000
TFIDF + MLP	0.56391480000
TFIDF + RF	0.58521970000
TFIDF + RF + LSTM（情感）	0.41877496000
TFIDF + ENSMBLE	0.60918770000

我们看到，因为 LSTM 模型在情感分类上过拟合严重，实际没有什么贡献（也有可能是我们的实现过于粗糙，模型健壮性不好），另外使用 LDA 特征向量的表现结果比使用 TFIDF 的效果差很多，可能是 LDA 模型不适用于这一问题的特征造成的，其余表现情况则是多层感知机模型强于一般的机器学习模型，随机森林模型更强，进一步集成的模型表现出更好的结果。

4.3 没有解决的问题和主要困难

1. 因为 LSTM 模型在对情感分类的学习中表现出严重的过拟合，如何通过采样特征表示得到新的样例
2. 对于 LDA 主题模型实验效果欠佳的解释
3. 将标签进行组合之后，每一类标签的实例数量更少，学习效果下降
4. 如何学习多主题分类的合理输出阈值

5 附录

5.1 个人收获

吴雨菲:

何秦兴:

1. **此次调研的最大收获是对** NLP **和机器学习有了基本的认识，了解到了不同的模型，也激起了我继续学习相关知识的兴趣。**
2. **感受到了理论和实践之间的巨大** Gap**，遇到了数据分析过程中一些很实际的问题，真切感受到了数据分析的一些困难，也激起了我对于利用数据科学解决这些困难的兴趣。**
3. **理解数据非常重要！恰当得处理数据比模型更重要。**
4. **感受到了动手能力的不足，要多锻炼这方面的能力！**

张劲暾:

1. **学习了关于** LDA **和** LSTM **模型的相关知识和相关机器学习框架的编程使用;**
2. **参与了一次线上正式比赛，意识到与优秀选手的差距;**
3. **积累了团队合作分工、机器学习工程构建的经验。**

5.2 分工说明

吴雨菲:

何秦兴:

1. LSTM **模型调研与编程实验**
2. **期末报告编写（**3.3.4 , 3.3.5, 3.3.6, 3.2.4 **部分)**

张劲暾: (**组长**)

1. **工作分配协调**
2. LDA **主题模型调研与编程实验**
3. **框架代码实现与方案具体实现，线上测试提交和比对调整参数**
4. **和何秦兴合作完成** LSTM **模型编程实现**
5. **期末报告编写（**1,2,4 **部分）排版**

6 参考文献

1. wikipedia:LSTM
2. 周志华:《机器学习》
3. Wenpeng Yin,Katharina Kann,Mo Yu, Hinrich Schutze.2017.Comparative Study of CNN and RNN for Natural Language Processing
4. Latent Dirichlet Allocation, May 2003, Journal of Machine Learning Research 3(4-5):993-1022, DOI: 10.1162/jmlr.2003.3.4-5.993
5. Sharing Clusters Among Related Groups:Hierarchical Dirichlet Processes
6. LDA-math - 文本建模
7. 文本主题模型之 LDA(一) LDA 基础
8. 文本主题模型之 LDA(二) LDA 求解之 Gibbs 采样算法
9. 文本主题模型之 LDA(三) LDA 求解之变分推断 EM 算法
10. 主题模型 TopicModel：Unigram、LSA、PLSA 模型
11. 主题模型 TopicModel：隐含狄利克雷分布 LDA
12. Dirichlet Process 和 Hierarchical Dirichlet Process
13. 分层 Dirichlet 过程（HDP）的理解
14. 层次狄利克雷过程（Hierarchical Dirichlet Processes)