KNN算法实现情感分类

作者：1818401002孙泽元

摘要：

情感分类是近年来国内外研究的热点，其任务是帮助用户快速获取、整理和分析相关评价信息，对带有情感色彩的主观性文本进行分析、处理、归纳和推理。自2000年初以来，情感分类已经成长为自然语言处理（NLP）中最活跃的研究领域之一。情感分类在现实生活中也有许多应用，如网上的商品评论，每条评论评价商品是表扬还是批评；再比如研究社交网络上网民对某一事物的评价积极的还是消极的，可以了解社会舆论；也有用社交网络网民发的twitter来与民意调查来进行对比的……关于文本分类有CNN，卷积神经网络，深度学习等算法，本文采用较为简单的KNN算法根据情感分类实现对文本的二分类。

关键词：

语料处理，向量表示，特征选择，KNN分类方法

正文：

首先介绍一下本次实验的大致思路：先对所给的四个语料集进行文本处理，使其便于向量表示，然后从处理好的训练集中选择特数目，缩小向量空间，提高分类效率，再从测试集中依次取出文本，计算其与训练集中语句的距离，找出与测试语句距离最近的k个训练语句，根据k个训练语句所属正例和负例的比例大小来判断测试语句所属的类。

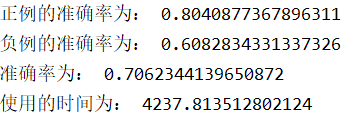
语料处理：英文中有大量单词没有实际意义，如“of”，“the”等，因此使用stopwords.words("english")将这些停用词去除。此外由于一个名词或动词可能有不同的形式，如“like”和“likes”，“film”和“films”这些单词表达的意思是相同的，因此使用WordNetLemmatizer().lemmatize()进行词形还原，PorterStemmer().stem()进行词干提取。有一点值得注意的是，上述两个步骤完成后得到的可能不是标准的英文单词，如“amaz”这样莫名奇妙的单词，不过这无伤大雅，因为这两步的目的就是将“amaze”，“amazing”这样意义相同的词统一成一个形式。为了便于调试，我在py程序目录下新建了三个txt文件train\_deal,test\_positive\_deal和test\_negative\_deal分别用来存储处理好的训练集（包括正例和负例），测试正例和测试负例，当向量表示时直接读取这三个文件避免每次运行都要重复进行语料处理，同时也便于我人工观察文本的特征。经观察发现，除了标点，一些数字出现频率较高，推测可能与所评价的电影或书籍内容有关，但其对情感分类没有帮助，因此利用isalpha()将其过滤（n’t较为特殊，先将其变为not避免被过滤）。关于语料处理还有其他手段如词性标注，然而考虑到有些词可能有多种词性，以及我无法确定某一词性对情感分类是否有帮助，所以并未采用。

向量表示和特征选择：向量表示和特征选择是我本次实验花费时间最多的部分，这两个部分对于分类的效果和性能影响最大，方法也最多。向量表示中常用的权重有布尔权重或者tf-idf权重，我认为对于idf来说，它本身倾向于文本中频率小的词，而二分类的情感词并不会太多，因此表达情感的词词频应该较高，而许多被idf重视的生僻词常常是无用的，使得tf-idf精度不高。经过测试也发现两种权重准确率相差无几，而tf-idf无疑会花费更多时间，因此我在本次实验中选择了布尔权重。关于特征数目的选择我用了最直接的词频，创建一个列表将经过语料处理的训练集中所有出现过的单词都放入其中，然后利用Counter函数进行统计，查看结果，发现出现次数极少的一些词都是与情感无关的生僻词，将其过滤，经测试准确率和性能都有较大提升。又发现出现次数最多的“film”“one”“movie”等词似乎对情感分类也没有意义，然而出乎我意料的是当我过滤了几个词之后准确率不升反降，经过分析推测可能是由于喜欢某部电影的人更倾向于在评论中多次提及，如“I like the film，the film impressed me a lot.”，而反之“I hate it，it’s boring.”当然这一推测没有什么理论和实验依据，仅仅是根据实验结果的好坏所作的猜想。这也反映了情感分类的一个困难：很多时候感性的语言难以用理性的数据精确地衡量。

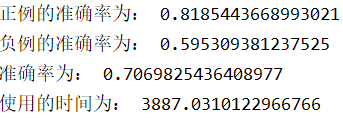
KNN算法：距离的选择是一个关键点，起初我用的是最常见的欧式距离，然而效果并不理想，经过反复测试最终选择了余弦距离（1-两向量夹角的余弦值），我认为在这一实验中，欧氏距离的缺陷在于计算距离的时候更注重单一的离散的点对距离的影响，而余弦距离更能反映一个整体的语句对距离的影响。例如点（1，1）和（1，0），根据欧氏距离结果是1，根据余弦距离结果是1-2\*\*（-1/2），显然前者更注重坐标第二个参数1和0对于距离的影响，而余弦距离显然考虑到了语句整体的构成。而用1减去夹角余弦值的原因是考虑到余弦值越大说明距离越小如相同向量夹角余弦值为1距离为0，并且同时使向量中所有数的取值都为0到1，实现了数据标准化。为统计距离最小的k个点所属正例还是负例，我创建了三个列表，一个列表为[1 for i in range(0,2995)] + [-1 for i in range(0,2995)]，1代表下标为0到2994的语句属于测试正例，-1表示下标为2995到5989的语句属于测试负例，另一个为空列表distances用来存储距离，还有一个top列表用来判断。从测试集中取出一条语句，计算其和训练集5990条语句的距离，存入distances列表，然后升序排列，取前k个距离，检索它们在5990个训练语句中的位置，从而在分类列表中得到1或者-1的值，存入top，最后对top求和。如果top和为正则判为正例，和为负则判为负例。然而这样做有两个问题：1.top为0难以判断为正例还是负例。2.每个距离对和的贡献一样，导致比如k为9，距离最近的4个点为训练正例，第5到第9近的5个点为训练负例，必然判为负例，这样的判断并不合理。因此在将1和-1放入top列表时我增加了一个权重：距离分之一，将其改为距离分之1或者距离分之-1，越近的语句贡献越大，同时增加权重之后top的和几乎不会为0，有效解决了上述两个问题。

测试不同特征数目N和KNN的K值对分类效果的影响：N对于分类效果和性能影响都较大，k在一定范围内对分类效果较小，对性能几乎无影响。

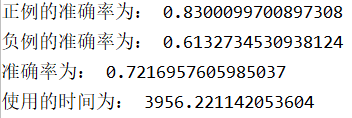
K=6，N=2470



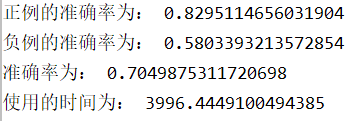
K=7，N=2470



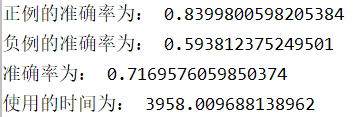
K=8，N=2470



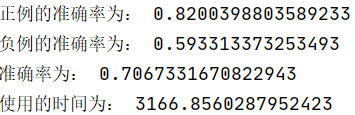
K=9，N=2470



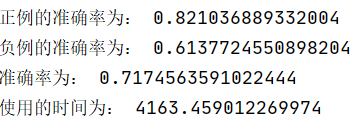
K=10，N=2470



K=8，N=2000



K=8，N=3000



综合来看，K=8，N=2470是一个近似最优解。

解释一下2470这个特征数目是如何取的：参考向量表示和特征选择这一部分我的思路，需要舍去一些生僻词和对分类效果影响不大的词。那么如果定义这些生僻词呢？我对处理过的训练集中的所有单词进行统计，发现出现次数最多的film出现了7995次，但film这个词不能明确体现情感，继续找到出现次数最多的明确表示情感的词like出现了4534次，于是认为出现次数小于等于45次（不到4354的百分之一）的词对于分类帮助不大将其过滤，最后的测试结果也表明过滤之后的准确率较高，性能相比未过滤时有巨大提升。若特征数目太小可能会漏掉更多对情感分类有影响的词，降低准确率；而特征数目太大则会大大增加运算量，导致巨大的时间开销。

以上内容就是本次实验的具体步骤。

小节：

平均准确率0.72，正例的准确率为0.83，这是一个不错的数据，然后负例不到0.62且无论我如何调整N和K，甚至是距离和权重都难以突破。我认为可能的原因有1.负例语料本身质量不高2.KNN算法的局限性3.还有更优的数据处理方法。

对于整个实验整体我最深的体会就是以前我们在编程时往往会忽视时间复杂度，然而在这个问题中时间成本成为了一个重要因素（多次测试才能得出合适的N和K），我仅仅在计算距离时减少了一层循环就将时间减少到之前的十分之一，让我更深地体会到细节决定成败。如果排除考试月忙碌地因素，这种考核方式还是非常有意思的，在动手实践中不仅深刻掌握了相关知识同时也能自己创新出一些独特的想法。

从神经网络的角度来看，依靠我现有的知识很难在0.72准确率的基础上实现大的提升，还有更多的知识需要学习。

参考文献：（以csdn博客为主）

<https://blog.csdn.net/qq_44271651/article/details/102536030>

<https://blog.csdn.net/weixin_41792682/article/details/89705971>

<https://www.jianshu.com/p/bfc41e537126>

<https://blog.csdn.net/zhaomengszu/article/details/81452907>

<https://blog.csdn.net/lucky_kai/article/details/89514868>

<https://blog.csdn.net/weixin_40547993/article/details/89485455>

程序说明：

语料和程序在同一目录下，其中3个后缀为\_deal的txt文件为新建的空txt文件，运行txt\_deal.py程序对语料加工处理,3个新建的txt文件分别存储处理后的训练集，测试正例和测试负例，然后运行array\_vector.py程序。这样做的原因是语料加工处理的过程较为固定，无非去停用词去符号词形还原词干提取，不需要每次调参之后再重新处理一遍，另外便于我查看文本处理之后的结果。最后由于考试月还要交两篇报告，KNN和特征数目没来得及写成函数的形式，如果要调整K需要在代码中改一处，调整N需要在代码中改7处（已用注释标出），其余细节也用注释标出。