货物智能分类

宋琰[[1]](#footnote-1)

1. 算法描述

1.1 用到的python库：jieba，sklearn，pandas，numpy，matplotlib

1.2 首先对分类器调参，然后比较几个分类器的准确率得分（训练集/测试集）

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | 朴素贝叶斯 | SGD Classifier |
| 三级分类 | 99.6/55.14 | 99.6/54.1 | 99.6/54.1 |
| 二级分类 | 83.0/83.5 | 82.7/83.4 | 82.6/83.3 |
| 一级分类 | 99.6/99.3 | 97.1/96.9 | 99.57/99.4 |

由于该问题为一个三级分类问题，经过多次实验测试，发现先对货物名称进行三级分类，而后根据三级分类进行二级分类，再根据二级分类进行一级分类。这样去进行分类的准确度最高。

**步骤：**

Step1：导入文件（train\_data.csv）,把货物分割成训练集和测试集。

Step2：对分类列使用jieba进行分词。

Step3：构建TF-IDF词向量空间

Step4：训练分类器，参数调优

Step4：导入待分类货物名称，先预测三级分类，结果保存至Dataframe。

再用预测好的三级分类作为训练集预测二级分类，保存至Dataframe，以此类推。

Step5：输出预测结果，保存为csv文件

1. 算法内容

编码工具为PyCharm，编码语言Python。

**Step1：**导入数据，按列保存至列表train。

**def** readtrain(path):  
 content\_train=pd.read\_csv(filepath\_or\_buffer=path,sep=**','**)[**"货物全称"**].values  
 level1\_train=pd.read\_csv(filepath\_or\_buffer=path,sep=**','**)[**'一级分类'**].values  
 level2\_train = pd.read\_csv(filepath\_or\_buffer=path, sep=**','**)[**'二级分类'**].values  
 level3\_train = pd.read\_csv(filepath\_or\_buffer=path, sep=**','**)[**'三级分类'**].values  
 train=[content\_train,level1\_train,level2\_train,level3\_train]  
 **return** train

**Step2：**对需分类列使用jieba进行中文分词。

**def** segmentword(cont):  
 c=[]  
 **for** i **in** cont:  
 a=list(jieba.cut(i))  
 b=**''**.join(a)  
 c.append(b)  
 **return** c

**Step3：**特征工程，用python自带的sklearn库对训练样本构建TF-IDF词向量空间。

train\_content,test\_content,train\_lable,test\_lable=train\_test\_split(content,lable,test\_size=0.4,random\_state=0)

先用train\_test\_split将样本切分，test\_size=0.4，即60%为训练样本，40为测试样本。输出分别为训练样本集，训练样本标签，测试样本集，测试样本标签。

vetorizer=CountVectorizer()  
tfidtransformer=TfidfTransformer()  
train\_tfidf=tfidtransformer.fit\_transform(vetorizer.fit\_transform(train\_content))

将训练样本集向量化，形成TF-IDF词向量空间。

Test\_tfidf=tfidtransformer.transform(vetorizer.transform(test\_content))

生成测试样本TF-IDF词向量空间。

**Step4：**训练分类器，参数调优。

将上一步分好的样本代入分类器训练，调整参数。

cls=MultinomialNB(alpha=0.01)  
cls=cls.fit(train\_tfidf, train\_lable)#构建贝叶斯分类器并进行训练

cls = SGDClassifier()  
cls = cls.fit(train\_tfidf, train\_lable)#构建SGDclassifier并进行训练

cls=SVC()

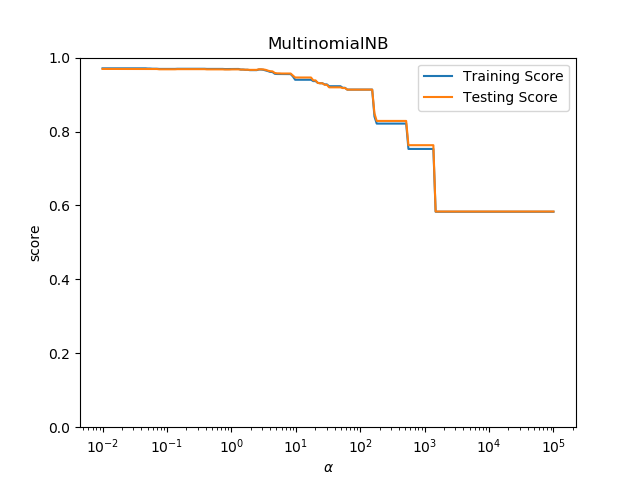
cls=cls.fit(train\_tfidf, train\_lable)#构建SVM分类器并进行训练

参数循环调优，以朴素贝叶斯为例。

alphas=np.logspace(-2,5,num=200)#在0.01和100000生产200个呈等比的数、

将这些数代入分类器，画出准确率-alpha图，代码和图如下：

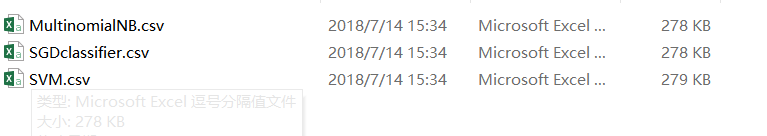
train\_scores = []  
test\_scores = []  
**for** alpha **in** alphas:  
 cls = MultinomialNB(alpha=alpha)  
 cls.fit(tfid1, train\_lable)  
 train\_scores.append(cls.score(tfid1, train\_lable))  
 test\_scores.append(cls.score(new\_tfid, test\_lable))  
fig=plt.figure()  
ax=fig.add\_subplot(1,1,1)  
ax.plot(alphas,train\_scores,label=**u"Training Score"**)  
ax.plot(alphas,test\_scores,label=**u"Testing Score"**)  
plt.legend()  
ax.set\_xlabel(**r"$\alpha$"**)  
ax.set\_ylabel(**"score"**)  
ax.set\_ylim(0,1.0)  
ax.set\_title(**"MultinomialNB"**)  
ax.set\_xscale(**"log"**)  
plt.show()

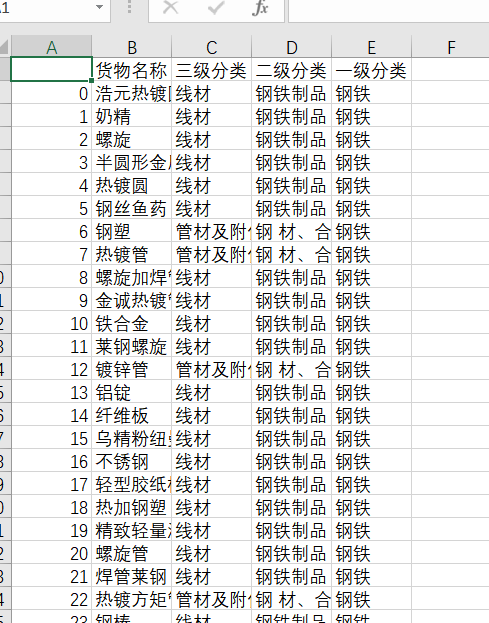


发现alpha最优值为0.01。对各分类器进行如上操作，发现最优参数即为默认参数。接着便将待分类货物导入，生成结果。

1. 分类结果

三个分类器生成了三个结果，分别为三个csv文件，文件从左至右，分别为货物名称，三，二，一级分类。见结果文件夹。





1. 宋琰（1995-），男，硕士研究生，研究方向：智能优化算法 通讯邮箱：[291873349@qq.com](mailto:291873349@qq.com)，手机：13701469711 [↑](#footnote-ref-1)