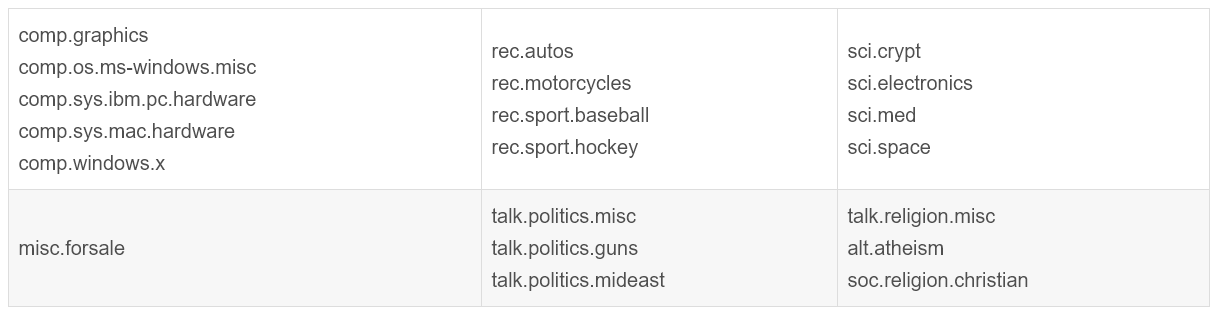
**高级机器学习——分类实验报告**

吕芳锐 高瓴人工智能学院 2020101273

1. **数据集**

实验的数据集选用20newsgroups数据集。20newsgroups数据集是用于文本分类、文本挖据和信息检索研究的国际标准数据集之一。数据集收集了大约20,000左右的新闻组文档，均匀分为20个不同主题的新闻组集合。一些新闻组的主题特别相似(e.g. comp.sys.ibm.pc.hardware/ comp.sys.mac.hardware)，还有一些却完全不相关 (e.g misc.forsale /soc.religion.christian)。其类别如下表所示：



在sklearn中，该模型有两种装载方式，第一种是fetch\_20newsgroups，返回一个可以被文本特征提取器，是可以自定义参数提取特征的原始文本序列；第二种是fetch\_20newsgroups\_vectorized，返回一个已提取特征的文本序列，即不需要使用特征提取器。

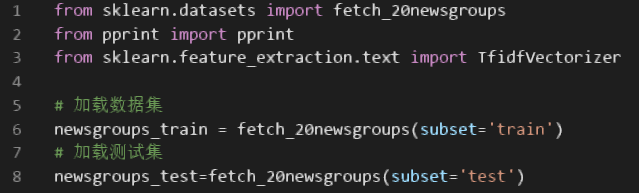
本次实验选取全部的原始文本序列，全部的20个类别，再使用TF-IDF统计词频，抽取特征。

TF-IDF是一种用于信息检索与文本挖掘的常用加权技术。TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF-IDF的主要思想是：如果某个单词在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

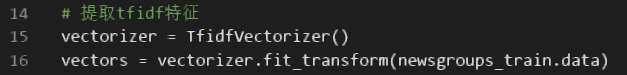
再抽取特征后使用具体的机器学习方法进行分类。

下面简单介绍处理数据集的流程：

首先加载数据集：



再提取TF-IDF特征向量：



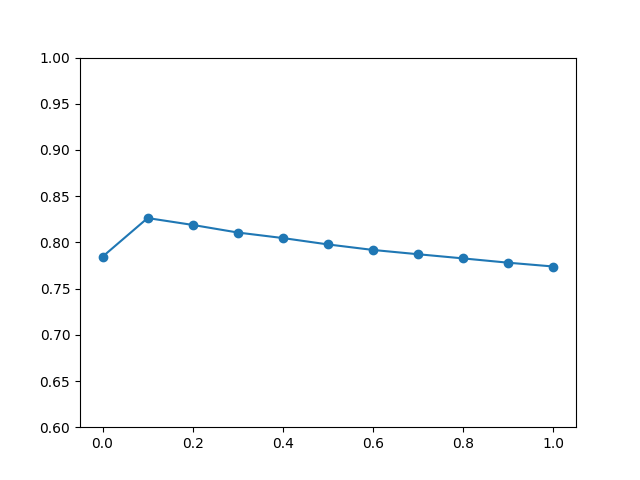
至此，数据处理已经完成，下面我们采用朴素贝叶斯、SVG、随机森林三种方法进行分类任务。下面将一一介绍三种方法以及实验结果。

1. **先验为多项式分布的朴素贝叶斯**

首先第一个实验使用的方法是贝叶斯方法。贝叶斯方法是以贝叶斯原理为基础，使用概率统计的知识对样本数据集进行分类。由于其有着坚实的数学基础，贝叶斯分类算法的误判率是很低的。贝叶斯方法的特点是结合先验概率和后验概率，即避免了只使用先验概率的主观偏见，也避免了单独使用样本信息的过拟合现象。贝叶斯分类算法在数据集较大的情况下表现出较高的准确率，同时算法本身也比较简单。

而朴素贝叶斯方法是在[贝叶斯](https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF/1405899)算法的基础上进行了相应的简化，即假定给定目标值时属性之间相互条件独立。也就是说没有哪个属性变量对于决策结果来说占有着较大的比重，也没有哪个属性变量对于决策结果占有着较小的比重。虽然这个简化方式在一定程度上降低了贝叶斯分类算法的分类效果，但是在实际的应用场景中，极大地简化了贝叶斯方法的复杂性。

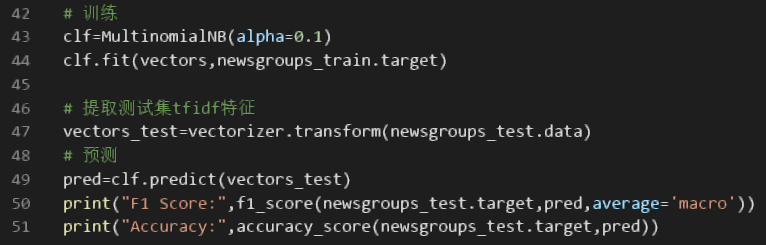
本次实验采用的是先验为多项式分布的朴素贝叶斯方法。在sklearn中，先验为多项式分布的朴素贝叶斯方法的函数为MultinomialNB(alpha=1.0，fit\_prior=True，class\_prior=None)。这个函数中有三个主要参数，分别代表：添加拉普拉斯平滑，是否要考虑先验概率，以及修改先验概率。在实验中发现第一个参数alpha对实验结果影响比较明显，下面给出算法准确率随alpha的变化：



可以看到alpha取到0.1时算法可以得到最好的效果，得到效果如下：



下面给出实验的代码：



1. **随机梯度下降(**SGD**)**

随机梯度下降（SGD）也称为增量梯度下降，是一种迭代方法，用于优化可微分目标函数。该方法通过在小批量数据上计算损失函数的梯度而迭代地更新权重与偏置项。SGD在高度非凸的损失表面上远远超越了朴素梯度下降法，这种简单的爬山法技术已经主导了现代的非凸优化。

SGD的惩罚方法penalty有三种设定方式：L1，L2，Elastic Net。其中elasticnet是L2 型和 L1 型的凸组合，下面我们一一尝试这项的设定：

|  |  |
| --- | --- |
| Penalty | Accuracy |
| L1 | 0.7774827403080191 |
| L2 | 0.8544875199150292 |
| Elastic Net | 0.8406797663303239 |

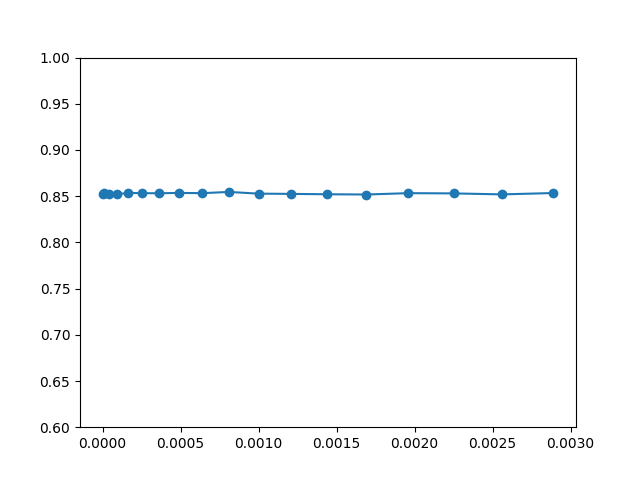
可以看到使用L1 penalty效果最差，这是因为使用L1 penalty导致稀疏解，使得大多数系数为零。 Elastic Net解决了在特征高相关时 L1 penalty的一些不足。Elastic Net中的参数 l1\_ratio 控制了 L1 和 L2 penalty的凸组合结合比例。可以看出L2的效果最好，我们应该使用L2方法进行训练。

SGD的损失函数可以设置为hinge（（软-间隔）线性支持向量机），modified\_huber（平滑的 hinge 损失），log（logistic 回归），和其他的回归损失。

|  |  |
| --- | --- |
| Loss Function | Accuracy |
| hinge | 0.8441332479915117 |
| modified\_huber | 0.8530270844397239 |
| log | 0.8264737121614445 |
| squared\_loss | 0.8360329261816251 |

这个设置对不同任务有着不同的效果。SGDClassifier 通过利用 “one versus all” （OVA）方法来组合多个二分类器，从而实现多分类。对于每一个 K 类, 可以训练一个二分类器来区分自身和其他 K-1 个类。SGD更适合执行二分类或者one vs all类型的任务，对于这个多分类任务效果不是太好。

下面是对于alpha参数的设置：

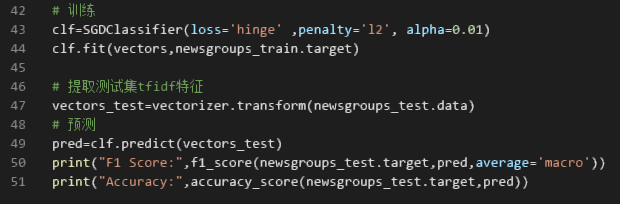


可见本任务对于alpha参数设置不敏感。

所以，最后我们选用Penalty为L2，Loss Function为hinge，得到以下结果：



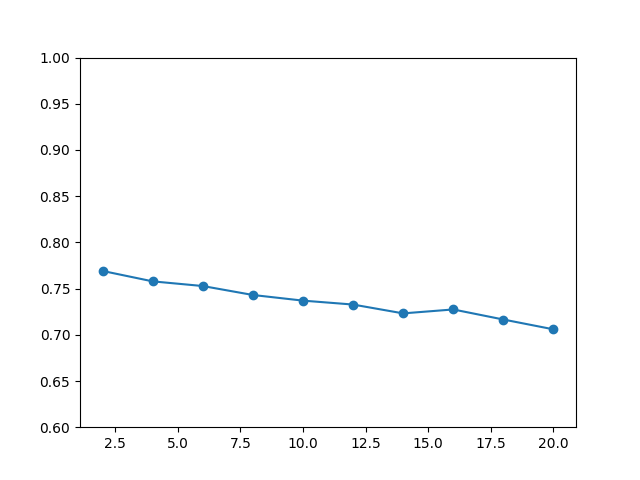
代码实现如下：



1. **随机森林**

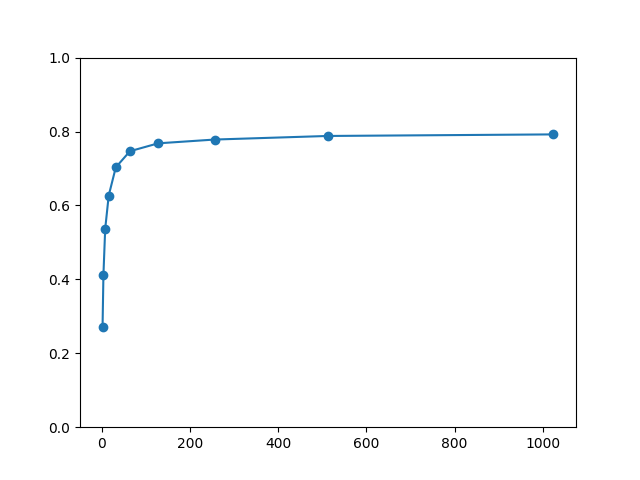
随机森林是由很多决策树构成的，不同决策树之间没有关联。当我们进行分类任务时，新的输入样本进入，就让森林中的每一棵决策树分别进行判断和分类，每个决策树会得到一个自己的分类结果，决策树的分类结果中哪一个分类最多，那么随机森林就会把这个结果当做最终的结果。

随机森林有关于树的最大深度的指标，min\_samples\_leaf，超过最大深度的树枝都会被减掉，是一个对分类结果非常重要的指标，下面我们看一下这个指标和精确度之间的关系。



由于实验问题数据规模不大，类别也不太多，随机森林在深度比较小时效果最好。

n\_estimator是树林中树木的数量，基评估器的数量。该参数越大，模型的效果往往越好，但是达到一定程度以后，精确性开始不在上升，或者开始波动。该参数越大需要的计算量和内存也越大，训练时间也越长，需要在训练难度和模型效果之间取得平衡。下面看一下n\_estimator和精确度之间的关系：

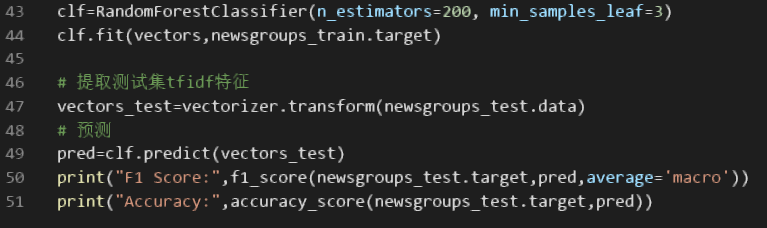


可见，当树林中树木的数量上升时，实验结果显著上升，但是当升高到200以上时随机森林数目对精确度没有太高的影响，和之前的分析一致。

那么，我们可以进一步优化算法，将树林中树木的数量设置为200，最大深度的指标为3，可以得到下面的结果，是目前位置准确率最高的参数优化：



具体代码实现如下：



1. **算法对比**

在本次实验的三种算法中，在精确率上SGD效果最好，朴素贝叶斯其次，随机森林最差。在时间上，朴素贝叶斯训练最快，SGD其次，随机森林最慢。可见在这个问题中使用SGD效果较好。

**优缺点总结：**

在这几种算法中，朴素贝叶斯算法假设了数据集属性之间是相互独立的，因此算法的逻辑性十分简单，并且算法较为稳定，当数据呈现不同的特点时，朴素贝叶斯的分类性能不会有太大的差异。换句话说就是朴素贝叶斯算法的健壮性比较好，对于不同类型的数据集不会呈现出太大的差异性。当数据集属性之间的关系相对比较独立时，朴素贝叶斯分类算法会有较好的效果。而属性独立性的条件同时也是朴素贝叶斯分类器的不足之处。数据集属性的独立性在很多情况下是很难满足的，因为数据集的属性之间往往都存在着相互关联，如果在分类过程中出现这种问题，会导致分类的效果大大降低。

SGD算法的最大优势在于易于实现，并且有很高的效率。但是也存在一些问题，它需要一些超参数，需要比较多的尝试以及人工调参经验。并且他对特征归一化是敏感的，不同的特征归一化可能带来非常不同的结果。

随机森林算法由于采用了集成算法，本身精度比大多数单个算法要好，所以准确性高。由于两个随机性的引入，使得随机森林不容易陷入过拟合（样本随机，特征随机）。在工业上，由于两个随机性的引入，使得随机森林具有一定的抗噪声能力，对比其他算法具有一定优势。它能够处理很高维度（feature很多）的数据，并且不用做特征选择，对数据集的适应能力强：既能处理离散型数据，也能处理连续型数据。在训练过程中，随机森林能够检测到feature间的互相影响，且可以得出feature的重要性，具有一定参考意义。但当随机森林中的决策树个数很多时，训练时需要的空间和时间会比较大。随机森林也有一些不好解释的地方，算是一个黑盒模型。同时在某些噪声较大的样本集上，随机森林容易陷入过拟合。

此外，我还运行了与其他算法的对比实验。但是因为使用全部的20个类别进行分类的话，有一些算法（比如SVM）训练时间过长，实验只采用挑选过的8个类别进行分类，具体实验结果如下：

|  |  |  |
| --- | --- | --- |
| 方法 | Accuracy | Time |
| 朴素贝叶斯 | 0.89602 | 2.297027 |
| SGD | 0.97339 | 2.384997 |
| Logistic回归 | 0.940519 | 4.440175 |
| SVC | 0.974284 | 38.494 |
| LinearSVC | 0.974955 | 2.780999 |
| LinearSVR | 0 | 2.25102 |
| MLP | 0.97585 | 184.125 |
| KNN | 0.459526 | 4.110004 |
| RandomForest | 0.9678 | 2.984966 |
| GradientBoosting | 0.921064 | 87.58929 |
| AdaBoostClassifier | 0.591682 | 5.659883 |
| DecisionTree | 0.97585 | 4.102994 |