hw2实验报告

52205903005 娄泽华

**hw2要求：**

数据集：<http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html>

任务：1000个文档分成20类，五重交叉验证结果，不要使用网站上的代码

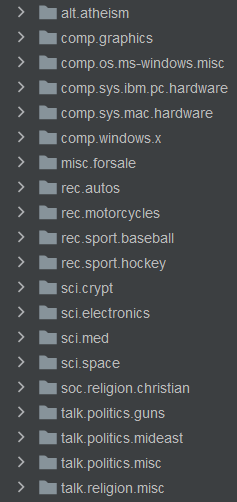
源码+实验报告

交给助教

Deadline: 学期末考试前

# 一、数据集简介

此数据集包含20000条新闻组消息的数据集，这些消息来自20个新闻组，每个新闻组的1000个文档。20个新闻组如下图：



# 二、实验思路

朴素贝叶斯分类器是最成功的文本分类算法之一。数据集包含20个类，每个类有1000个实例，其中每个类表示一个新闻组。使用朴素贝叶斯分类器实现文本分类，先读入文本，转换为词频张量，打乱顺序，用TF-IDF转换为词频-逆词频特征，最后用朴素贝叶斯分类器预测并进行5折交叉验证检验模型的性能。

# 三、代码实现

## 3.1 预处理

读入文本，转换为转换为词袋矩阵

def preprocess(news\_groups\_dir):  
 X = []  
 y = []  
 news\_groups = os.listdir(news\_groups\_dir)  
 for news\_group in news\_groups:  
 news\_group\_path = os.path.join(news\_groups\_dir, news\_group)  
 news\_names = os.listdir(news\_group\_path)  
 for news\_name in news\_names:  
 with open(os.path.join(news\_group\_path, news\_name), 'rb') as f:  
 news\_raw = f.read().decode('ANSI', 'ignore')  
 words = nltk.tokenize.word\_tokenize(news\_raw)  
 X.append(words)  
 y.append(news\_group)  
 return X, y

打乱顺序并转换为一一对应的x,y完整数据对：

data\_class\_zip = list(zip(X, y)) # zip压缩合并，将数据与标签对应压缩  
random.shuffle(data\_class\_zip) # 将data\_class\_list乱序  
X, y = zip(\*data\_class\_zip)

# 转换为array  
x = []  
for i in range(len(X)):  
 x.append(' '.join(X[i]))  
x = np.array(x)  
y = np.array(y)

TF-IDF及NaiveBayesian算法的pipline，其中拉普拉斯平滑参数取0.2：

# process pipline  
nbc = Pipeline([  
 ('vect', TfidfVectorizer(stop\_words=['\u3000', '\x00'], ngram\_range=(1, 2), max\_df=0.5  
 )),  
 ('clf', MultinomialNB(alpha=0.2, fit\_prior=True)), # alpha: Additive (Laplace/Lidstone) smoothing parameter.  
])

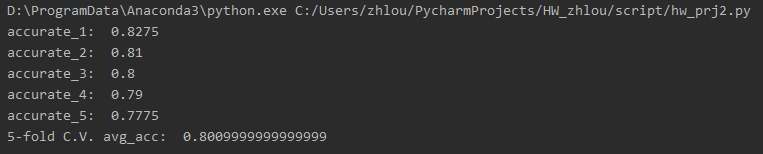
## 3.2训练、预测、K折交叉验证

K\_fold = 5时，即为5折交叉验证：

# K-fold cross validation  
avg\_acc = 0  
i = 1  
kf = KFold(n\_splits=K\_fold)  
for train\_index, test\_index in kf.split(x):  
 x\_train, y\_train = x[train\_index], y[train\_index]  
 x\_test, y\_test = x[test\_index], y[test\_index]  
 nbc.fit(x\_train, y\_train) # 训练我们的多项式模型贝叶斯分类器  
 y\_predict = nbc.predict(x\_test) # 在测试集上预测结果  
 y\_predict = np.array(y\_predict)  
 y\_test = np.array(y\_test)  
 acc = np.mean(y\_predict == y\_test)  
 print('accurate\_{}: '.format(i), acc)  
 avg\_acc += acc  
 i += 1  
avg\_acc /= K\_fold  
print('{}-fold C.V. avg\_acc: '.format(K\_fold), avg\_acc)

# 四、实验结果

5折交叉验证后的准确率为80.100%。



# 五、实验心得

本次实验中，我在编程过程中了解了pipline的使用，并熟悉了朴素贝叶斯的原理，尤其是对从文本到词典词频矩阵到TF-IDF的批量转换等预处理有了更清晰地认识。此次实验的关键点主要在于对数据的预处理，对朴素贝叶斯及交叉验证的理解上，我用了多项式朴素贝叶斯模型，并学习其先验：虽然初始每个类别的文本个数相同，但在交叉验证时划分了验证集合测试集之后还是需要学习先验的。