数据挖掘实验报告

学院：计算机科学与技术学院
专业：计算机技术 2018 研
姓名：冷汉超
学号：201834868

2018/12/31

1 VSM and KNN

I. 实验任务

A. 预处理文本数据集, 并且得到每个文本的 VSM 表示;

B. 实现 KNN 分类器, 测试其在 20Newsgroups 上的效果。

II. 实验方法

A. 文本预处理，向量化，根据特征词的 TF*IDF 值计算;

B. 当新文本到达后，根据特征词计算新文本的向量;

C. 在训练文本中选出与新文本最相近的 K 个文本，相似度用向量夹角的余弦值度量;

D. 在新文本的 K 个相似文本中，依此计算每个类的权重，每个类的权重等于 K 个文本中属于该类的训练样本与测试样本的相似度之和;

E. 比较类的权重，将文本分到权重最大那个类别中。

III. 实验过程

A. 分词

读取文档按空格分词, 并且去掉符号。将文档划分成单词, 并对单词做一些处理: 大写字母变成小写字母, 名词复数变单数, 去掉停用词, 各种时态和形式的动词变成原形, 只保留英文词干部分。

B. 划分训练集和测试集

将数据集划分成训练集和测试集, 其中训练集占 80%, 测试集占 20%。

C. 创建词典

从训练集中读取所有的文档, 统计所有的单词及词频, 计算 TF-IDF 的值, 提取关键词, 并创建字典。

D. 使用 KNN 进行文本分类

计算每一个测试实例到训练集实例的欧式距离, 对所有距离进行排序, 得到 K 个最近邻。对最近邻进行合并排序, 最后测试分类准确度。

IV. 实验结果

```
['years', 'yeast', 'yes', 'yesterday', 'york', 'young', 'zero', 'zip', 'zone', 'zoo']
['years', 'yeast', 'yes', 'yesterday', 'york', 'young', 'zero', 'zip', 'zone', 'zoo']
(15062, 3000)
58.357190280175274
(3766, 3000)
55.773234200743495
predict info:
precision:0.792
recall:0.792
f1-score:0.792
```

V. 实验总结

KNN 的分类方法因为没有训练过程，所以，分类时特别慢，因为需要和所有样本进行比较，同时特征维数很多，也要影响效率。分类准确度还可以，基本和没有优化过的 svm 差不多。kNN 的准确度和样本数量还是蛮相关的，kNN 因为效率的问题，所以在实际中应用还需要慎重。

2 NBC

I. 实验任务

使用朴素贝叶斯分类器, 测试其在 20 Newsgroups 数据集上的效果

II. 实验方法

A. 文本预处理，向量化，根据特征词的 TF*IDF 值计算；

B. 当新文本到达后，根据特征词计算新文本的向量；

C. 比较测试集词典和训练集词典, 如果有相同的词, 则对应词的 p(每个词 | 每个类) 可以从训练集中延用, 可以放入测试集概率向量列表中；

D. 将测试集中词典的所有 p(每个词 | 每个类) 相乘再乘以类概率 pclass 得到该文档属于此类的概率, 依照此方法计算出每个测试文档属于各个类的概率；

E. 比较每个训练文档属于各个类的概率, 找到最大的概率, 并将文档归于相应的。类。

III. 实验过程

A. 分词

读取文档按空格分词, 并且去掉符号。将文档划分成单词, 并对单词做一些处理: 大写字母变成小写字母, 名词复数变单数, 去掉停用词, 各种时态和形式的动词变成原形, 只保留英文词干部分。

B. 划分训练集和测试集

将数据集划分成训练集和测试集, 其中训练集占 80%, 测试集占 20%。

C. 创建词典

从训练集中读取所有的文档, 统计所有的单词及词频, 计算 TF-IDF 的值, 提取关键词, 并创建字典。

D. 使用 NBC 进行文本分类

由于使用词的出现次数作为特征, 可以用多项分布来描述这一特征。在 sklearn 中使用 sklearn.naive_bayes 模块的 MultinomialNB 类来构建分类器。使用 Pipeline 这个类来构建包含量化器 (vectorizers) 和分类器的复合分类器 (compound classifer)。

IV. 实验结果

```
['years', 'yeast', 'yes', 'yesterday', 'york', 'young', 'zero', 'zip', 'zone', 'zoo']
['years', 'yeast', 'yes', 'yesterday', 'york', 'young', 'zero', 'zip', 'zone', 'zoo']
(15062, 3000)
58.10330633382021
(3766, 3000)
56.37095061072756
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
predict info:
precision:0.830
recall:0.830
f1-score:0.830
```

V. 实验总结

朴素贝叶斯用于很多方面，数据有连续和离散的，连续型时可用正态分布，还可用区间，将数据的各属性分成几个区间段进行概率计算，测试时看其属性的值在哪个区间就用哪个条件概率。再有 TF、TDIDF，这些只是描述事物属性时的不同计算方法，例如文本分类时，可以用单词在本文档中出现的次数描述一个文档，可以用出现还是没出现即 0 和 1 来描述，还可以用单词在本类文档中出现的次数与这个单词在剩余类出现的次数（降低此属性对某类的重要性）相结合来表述。

3 Clustering with sklearn

I. 实验任务

测试 sklearn 中聚类算法在 tweets 数据集上的聚类效果。并使用 NMI(Normalized Mutual Information) 作为评价指标。

II. 实验方法

A. 调用 KMens 方法, 返回预测的聚类结果;

B. 使用 NMI 评价标准, 将真实聚类结果与预测值作比较, 得到 NMI 值;

C. 调用 Affinity propagation 方法返回聚类结果并计算 NMI 值;

D. 调用 Mean-shift 方法返回聚类结果并计算 NMI 值;

E. 调用 Spectral clustering 方法返回聚类结果并计算 NMI 值;

F. 调用 Ward hierarchical clustering 方法返回聚类结果并计算 NMI 值;

G. 调用 Agglomerative clustering 方法返回聚类结果并计算 NMI 值;

H. 调用 DBSCAN 方法返回聚类结果并计算 NMI 值;

I. 调用 Gaussian mixtures 方法返回聚类结果并计算 NMI 值。

III. 实验过程

A. 读取数据，使用 json 格式解析

B. 分别调用 8 种聚类方法

C. 使用 NMI 评价指标分别评价 8 次不同聚类方法所得到的聚类效果

IV. 实验结果

```
# num_cluster: 110
# K-means NMI: 0.7916051205577013
# AffinityPropagation NMI: 0.7834777200368183
# MeanShift NMI: 0.7468492000608157
# SpectralClustering NMI: 0.6716412603878753
# AgglomerativeClustering NMI: 0.7758740356993199
# DBSCAN NMI: 0.7009526046894612
# GaussianMixture NMI: 0.7859487200756089
```

V. 实验总结

K-means 算法是是最经典的聚类算法之一，它的优美简单、快速高效被广泛使用。它是很典型的基于距离的聚类算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把得到紧凑且独立的簇作为最终目标。