Journal of System Simulation

Volume 31 | Issue 12 Article 20

12-13-2019

Noise Clipping Algorithm Based on Relative Contribution Rate

Shuoyu Liu
School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China;

Yueming Dai
School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China;

Noise Clipping Algorithm Based on Relative Contribution Rate

Abstract

Abstract: This paper presents a class noise cutting algorithm (Class noise cutting, CNC) based on relative contribution rate. *The algorithm calculates the relative contribution rate of features to the theme. The most valuable feature set is selected by using features distinguish rating. The corresponding candidate categories for each feature are selected*, to reduece the candidate category set, improves the classification accuracy, and speed up the response speed of the classifier. Compared with another ECN noise cutting algorithm (Eliminating the class whose), CNC-has higher accuracy and because of its simpler feature dimension dictionary and better candidate category set, the response speed is greatly accelerated.

Keywords

relative contribution rate, class noise cutting, hierarchical classification, feature selection

Recommended Citation

Liu Shuoyu, Dai Yueming. Noise Clipping Algorithm Based on Relative Contribution Rate[J]. Journal of System Simulation, 2019, 31(12): 2721-2730.

第 31 卷第 12 期 系统仿真学报© Vol. 31 No. 12
2019 年 12 月 Journal of System Simulation Dec., 2019

基于相对贡献率的噪声裁剪算法

刘朔瑜，戴月明
(江南大学 物联网工程学院，江苏 无锡 214122)

摘要：提出了一种基于相对贡献率的噪声裁剪算法(Class noise cutting, CNC)。*通过计算得到特征对于主题的相对贡献率，利用特征区分度评分挑选对于当前主题分类最有价值的特征集，选出相应的候选类别,减少候选类别集*，提高了分类准确率，加快了分类器的响应速度。与另一种噪声裁剪算法(Eliminating class noise, ECN)比较，CNC 具有更高的准确率，由于具有更精简的特征维度词典以及更优异的候选类别集使得响应速度大大加快。

关键词：相对贡献率；类别噪声裁剪；层次结构分类；特征选择
中图分类号：TP278 文献标识码：A 文章编号：1004-731X (2019) 12-2721-10
DOI: 10.16182/j.issn1004731x.joss.19-FZ0289

Noise Clipping Algorithm Based on Relative Contribution Rate

Liu Shuoyu, Dai Yueming
(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: This paper presents a class noise cutting algorithm (Class noise cutting, CNC) based on relative contribution rate. *The algorithm calculates the relative contribution rate of features to the theme. The most valuable feature set is selected by using features distinguish rating. The corresponding candidate categories for each feature are selected*, to reduece the candidate category set, improves the classification accuracy, and speed up the response speed of the classifier. Compared with another ECN noise cutting algorithm (Eliminating the class whose), CNC-has higher accuracy and because of its simpler feature dimension dictionary and better candidate category set, the response speed is greatly accelerated.

Keywords: relative contribution rate; class noise cutting; hierarchical classification; feature selection

引言

进入 21 世纪，随着互联网的快速全球化，电子数据的数量急速增长，传统的人工分类由于准确率与效率的低下，很难适应大数据量的处理，文本分类作为解决方法得到了长足的发展。大规模层次分类[1]是指根据一个规模巨大的类别层次，指定未知类别对象在层次基于不同信息中所属的类别。分类的方法一般是机器学习中的自动分类方法，类别的层次可以是专家制定，也可以是分类过程中通过聚类学习产生。

其中分类作为数据处理的关键技术，国内外学者尤其对大规模分类问题尤为重视，继而进行了更深入系统的研究。国内的学者研究取得很大进展[2]综合近几年本领域的发展实际，在一般性的层次分类问题层面，总结了大规模多层次文本分类的概念和求解思路，并对传统的分类算法进行综合性的评价比较，并且在目前研究进展的基础上提出了 3 个大规模多层次文本分类的研究点，而且简述了

收稿日期：2019-04-16 修回日期：2019-07-07;
基金项目：国家自然科学基金(61973138);
作者简介：刘朔瑜(1993-)，男，江苏无锡，硕士生，研究方向为自然语言处理；戴月明(1968-)，男，江苏常熟，硕士，副教授，硕导，研究方向为自然语言处理，人工智能。

研究中的难点和重点。国外学者在这方面的研究起步比国内学者的要早，而且研究成果丰厚、方法实用有效、值得借鉴学习，例如，Sun 等尝试通过计算类别之间的相关距离，并结合类别相似性等前期信息来判断待分类文本的类别属性，此方法还可以判断类树层次结构的内部节点类别和叶子节点类别[3]；Liu 等从数学分析的角度出发，在大规模多层次分类中采用了具有层次式结构的 SVM 分类方法来减少分类过程所用的时间，数据结果显示此方法的学习时间是理想的，效果还优于一般的经典分类方法[4]。因此，对于大规模多层次文本分类问题，高精度的多层次文本分类技术非常有必要继续深入研究。

大规模层次分类[5]分为 3 种分类策略：全局处理(overall-conquer)、分而治之(divide-and-conquer)、化繁为简(reduce-and-conquer)。本文所提出的方法就属于化繁为简策略，也可以叫做收缩分类方法。它主要分为两个阶段：首先是根据待分类文档在所有给定的类别集合中搜索到与此文档相关的候选类别，去除掉噪声类别[6]；然后利用分类器对候选类别进行分类。这种方法减少了用于分类的类别噪声，一方面提高了准确率，另一方面也加快了响应速度。

1 类别噪声裁剪方法相关定义

1.1 相对贡献率

为了对相对贡献率进行定量分析，本文引入了以下一些概念：$W=(a_1,a_2,a_3,...a_n)$ 为一个样本文档，$B_k=(W_1^k,W_2^k,W_3^k,...W_m^k)(k=1,2,3,...z)$ 为一个样本集，k 表示类别，m 表示文本的数量。

在文本分类中，一个特征向量在类别中的出现频次代表了这个特征向量对于类别的重要性，而一个特征向量在不同类别中出现频次的差值则体现了这个特征词对于不同类别的区分度。假设特征词 $a_i(i=1,2,...,n)$ 在类别 k_1 中出现的频次为 t_1，在类别 k_2 中出现的频次为 t_2。如果 $t_1>t_2$，则代表特征词 a_i 对于类别 k_1 的重要性大于 k_2 的重要性。t_1-t_2 则表示该特征词在 2 个类别中的频次差，差值越大表示这个词越能很好的区分这两个类别，t_1、t_2 我们称作特征词的绝对频次，而 t_1-t_2 我们称作相对频次。为了防止相对频次出现负数影响之后的计算，本文的相对频次采取的是 t_1/t_2 的比值形式，他同样可以表示两个类别的区分度。

假设有一个特征词 a_1，在多类别分类的情况下，a_1 对任意两个类别的相对频次是不同的，我们选取所有的相对频次之和作为特征词 a_1 对于某一个类别的贡献率，即相对贡献率，见式(1)。

$$R(t_i,c_k)=\sum_{j=1,j\neq k}^{z}\frac{t_i^{\,k}}{t_i^{\,j}} \tag{1}$$

式中：$t_i^{\,k}$ 为特征词 a_i 在类别 c_k 中的绝对频次；$t_i^{\,j}$ 为特征词 a_i 在类别 j 中的绝对频次；z 为类别的总数；t_i 为特征词；c_k 为类别。

由于样本集的种类各异，可能一些是只有几百字的短文本文档，一些是有上千字的长文档，这时绝对频次和相对频次由于受到文本长度的影响不能很好的代表特征词对于类别的贡献率，所以本文采用特征词的相对频次与类别的所有特征词相对频次之和的比代表新的相对贡献率，见式(2)。

$$R(t_i,c_k)=\sum_{j=1,j\neq k}^{z}\frac{t_i^{\,k}/t^k}{t_i^{\,j}/t^j} \tag{2}$$

式中：t^k 为类别 c_k 中所有的特征词的相对频次之和；t^j 为类别，c^j 中所有的特征词的相对频次之和，其余的参数与公式(1)含义相同。公式(2)经过拉普拉斯平滑之后，变成了公式(3)：

$$R(t_i,c_k)=\sum_{j=1,j\neq k}^{z}\frac{(t_i^{\,k}+1)/(t^k+M)}{(t_i^{\,j}+1)/(t^j+M)} \tag{3}$$

式中：M 指的是类别，c_k 的每个特征词绝对频次加一之后，这个类别总体增加的绝对频次。

定义 1. $R(t_i,c_k)$ 是公式(3)对应的相对贡献率，则特征词 t_i 对类别 c_k 的贡献率 Ca_{ik} 为：

$$Ca_{ik}=\frac{\log(R(t_i,c_k)+1.0)}{\sqrt{\sum_{i=1}^{n}\left[\log(R(t_i,c_k)+1.0)\right]^2}} \tag{4}$$

式中：n 是指类别 c_k 中的特征词数量。

1.2 基于贡献率的类别区分度

定义 2. 上文中定义了特征词对于类别的贡献率，主要体现了特征词对于类别的重要性，为了进一步剔除掉对于当前主题分类没有太大影响的噪声特征，本文定义区分度评分如公式(5)所示：

$$DC(a_i)=\frac{\sqrt{\sum_k\left(Ca_{ik}-\overline{C_i}\right)^2}}{\sum_k Ca_{ik}}\qquad(5)$$

式中：Ca_{ik} 为 1.1 中的贡献率；$\overline{C_i}$ 为特征词 a_i 在各类别中贡献率的均值；z 为主题类别的总数。

然后，设置一个可以过滤掉噪声特征词的阈值 $LowestScore$，如果 $DC(a_i)>LowestScore$，则认为该特征对于分类有用，挑选出来；否则就是噪声特征，直接剔除。

公式(5)中区分度评分的核心是标准差公式，因为标准差可以很好的反应特征词集内个体特征词贡献率的离散程度。离散程度越高，表示此特征词对于不同类别的重要性差别越大，对于分类的价值越高。

1.3 特征的候选类别筛选

在章节 1.2 对重要特征进行提取之后，保留了一些对于分类有价值的特征，每个特征对于各个类别都有一个贡献率，但是有些贡献率特别低，表示特征词对于这个类别的区分没有太大的作用，所以本文设置一个阈值 $ScreeScore(ScreeScore>0)$ 来过滤掉那些和此特征词几乎没有联系的类别。

定义 3. 假设 a_i 是经过 1.2 提取之后的重要特征值，定义特征的候选类别集合为：

$$CS_i=\{c_k\,|\,Ca_{ik}>ScreeScore\}\qquad(6)$$

式中：Ca_{ik} 为特征词 t_i 对类别 c_k 的贡献率，阈值 $ScreeScore>0$。

$LowestScore$ 与 $Screescore$ 两个阈值的设定请详见章节 3.2。

1.4 回溯机制

在对测试样本经过上述步骤之后，得到了属于该测试样本的候选类别集，这个候选类别集是总类别的一个子集，而且多数情况下是一个真子集。如果该测试集所属真正的类别不在此候选类别集中，那么接下来的分类等等一系列操作都属于浪费时间，因此，我们制定了一个回溯机制，尽可能的保证将真正的类别包含在候选类别集中。

1.4.1 回溯机制触发条件

设定一个阈值 Trigger 用来检测初始候选类别集中是否有不满足条件的类别。

$$p(y=c_k\,|\,a)=p(y=c_k)\prod_i p(a_i|\,y=c_k)\qquad(7)$$

朴素贝叶斯分类器是一种准确率高、适应范围广、计算复杂度低的分类器。公式(7)是朴素贝叶斯的概率公式，它表示的是一个文档属于某个类别的概率。设定对于类别 c_k，经公式(7)计算出的值为 $CF_{c_k}(W)$ (W 是一个样本)。

定义 4. 对于某个类别 c_k，设 W 是训练集中属于该类别的一个文档，则有

$$Trigger(c_k)=\min\left\{CF_{c_k}(W)\right\}\qquad(8)$$

公式(8)表示在训练集中将类别 c_k 的所有样本经过公式(7)计算得到的 CF_{c_k} 的最小值作为此类别的阈值。因为当训练集规模比较大的时候，样本集中某类的数据经过公式(7)计算得到的 CF_{c_k} 最小值可以近似看成此类别文档属于该类的最小概率。当待分类文档的候选类别集中某个类别经过朴素贝叶斯公式计算得到的值小于该类的 Trigger 值，则认定此类别应该不是真实类别，将其从候选类别集中剔除。

1.4.2 最相似类别

公式(7)表示的是一个文档属于一个类别的概率，在章节 1.4.1 对所有文档以及所有类别经过该公式计算之后，我们发现很多文档计算出的最大概率并不是它的真实类别，这是朴素贝叶斯方法本身的局限性导致的，它并不能完美的预测所有文档，但是当某个类别的很多文档经过公式(7)计算之后

得到的最大概率都来自于另外一个类别时，那这两个类别一定有很高的相似性。

定义 5. 设 $A_i(j)$表示属于类别 c_i 的训练集文档与类别 c_j 经过公式(7)计算之后得到的概率值最大(即大于此文档属于其它类别的概率值)的文档数量。那么有：

$$MSC(c_j)=\{c_j \mid \max(A_i(j))\} \tag{9}$$

式中：c_i 为除了类别 c_j 之外的其他类别。公式(8)就表示 c_j 的最相似类别是 c_i。

经过公式(9)计算之后，每个类别会得到它的最相似类别。最相似类别主要作为当前被删除的候选类别的替补进入候选类别集中，进一步提高候选类别集的准确率。

1.4.3 回溯过程

当对一个测试文档 g 的初始候选类别集进行检测时，回溯机制条件触发，不符合条件的类别 c_j 被从候选类别集中剔除，并将此类别放入淘汰类别集中。观察此类别的最相似类别 c_l 有没有在淘汰类别集中，如果在说明已被淘汰，继续观察 c_l 的最相似类别；如果不在，说明 c_l 未被淘汰，检测 c_l 是否满足回溯触发条件，如果满足，就继续回溯，不满足，就将 c_l 放入候选类别集中。循环以上过程，直至候选类别集中的所有类别都通过回溯检测或者类别为空，如果类别为空，则从公式(9)得到的集合中再选择 m 个组成初始候选集进行回溯，但是这种情况几乎不可能发生。

2 算法裁剪过程及时间复杂度

2.1 算法裁剪过程

本文所提出的类别裁剪算法[7]属于层次分类方法中的收缩分类方法(Reduce Approachers)。收缩分类方法是一种先搜寻然后分类的算法，即首先通过某种算法搜索出与特征词联系紧密的类别，存储在该特征词的候选类别中，当对一篇测试文档进行分类时，直接从候选类别中选出该测试文档中所有特征词的候选类别构成一个候选类别集，然后利用分类器只针对候选集中的类别进行分类，如图 1 所示。

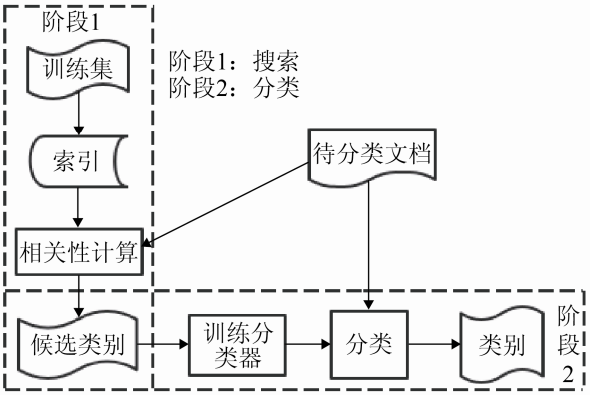

图 1 两阶段分类方法

Fig. 1 Two-stage classification method

它的主要原理是通过建立候选类别集使得待分类文档减少学习的主题数量来提高准确率以及加快响应速度。如图 1 中的两阶段的分类算法是默认待分类文档与给定的不止一个主题相关，因此更加适合多类别单标签分类[8-9]，且类别越多越好。

本文所提出的基于相对贡献率的类别噪声裁剪算法详细过程如下：

一．搜索阶段：

1. 对给定的训练集进行数据预处理；

2. 对预处理后的数据集依据公式(3)~(4)计算相对贡献率；

3. 对特征的相对贡献率计算出相应的类别区分度评分，并通过与 *LowestScore* 的对比过滤掉噪声特征；

4. 针对过滤后的特征集，通过与 *ScreeScore* 对比为每个特征筛选出对应的候选类别集合，并存储起来。

二．分类阶段：

1. 对给定的待分类文档进行数据预处理;

2. 从搜索阶段筛选出的特征集中找到该文档所包含的特征，存储该特征所对应的主题集合;

3. 利用分类器只针对主题集合中的类别进行分类;

4. 输出分类结果。

2.2 时间复杂度

输入：待分类文档集 $B=(W_1,W_2,W_3,...W_k)$

输出：B 的所属类别集

功能：通过 CNC(Class noise cutting)算法为 B_i 中的每个文档预测所属的主题

1. $Q=\varphi$； // 存储最终的分类结果集合；
2. $Y=\varphi$； // 一个文档搜索到的候选类别集合；
3. $P=\varphi$； // 所有文档搜索到的候选类别集合；
4. L； // 过滤后的特征词集合；
5. CS_i；// a_i 对应的候选类别集合；
6. *for each* W_i *in* B;
7. *for each* a_i *in* W_i;
8. *If* a_i *in* L;
9. $Y=Y\cup CS_i$;
10. $P=P\cup Y$;
11. *END*;
12. *for each Y in P*:
13. $Q=Classfied_W(Y)\cup Q$； // 利用分类器对候选集进行分类；
14. *OutPut Q*。

算法时间复杂度分析：本算法的时间复杂度主要集中于步骤(7)、(13)，步骤(7)是针对待分类文档中预处理后的特征词进行遍历并与特征词典比较存储候选类别集。由于待分类文档一般篇章长度有限，预处理后特征词数量很少，而且经过学习后的特征词典一般维度不是太大，所以可以忽略；所以，复杂度主要集中于步骤(13)中分类器的作用时间。如上所示，待分类文档集 B 一共有 k 篇文档，假设共有 x 个类别，如果没有进行类别噪声过滤，分类的时间复杂度为 $O(kx)$；在利用本算法进行类别筛选后，每篇文档最终分类的候选类别集合 $g<<x$ (该算法针对的一般是类别数目比较多的样本集)，分类的时间复杂度为 $O(kg)$，极大地提高了响应速度。

3 实验设置与结果分析

3.1 实验设置

本实验为了试验算法的适应性，同时选取了中英文样本集进行实验。中文样本集采用的是搜狗实验室中的全网新闻数据，共分为 20 个类目，每个类目有 2 000 篇文章，共 40 000 篇文章。英文样本集采用的是 Reuters 数据集 15 000 篇，训练集选取了 12 000 篇文档，测试集有 3 000 篇文档。

本实验的开发平台是 Inter(R) Core(TM) i5-4590CPU @3.30 GHZ 3.30 GHZ，安装内存为 4.00 GB，操作系统为 WIN7 64 位操作系统。开发环境是 python3.6，开发工具是 pycharm2016 和 Matlab2014。

实验中，分词工具采用的是 jieba 分词，权值计算采用的是 Okapi[10]计算公式，以朴素贝叶斯[11-12]与 SVM[13-14]分别作为分类器。其中，朴素贝叶斯分类器采用的是高斯核，SVM 采用的是一对多的线性分类器。实验评价指标采用的分别是宏平均 *Macro-F*。

为了测试 CNC 算法对于噪声类别的过滤效果以及对于分类效率和准确率的影响，本文分别安排了以下 2 组实验：

1) 多算法对比实验：在不同噪声类别比例的情况下，将 CNC 算法与多种特征选择方法在中英文数据集上进行对比实验，统计实验结果，画出图表。

2) 噪声裁剪算法总性能对比实验：用另一个性能优秀的类别裁剪算法 ECN 与 CNC 进行响应速度与准确率的对比实验，并分别选取两种算法的巅峰性能所对应的特征维度区间，画成图表。

3.2 参数设置

在定义 2 与定义 3 中有 2 个阈值没有设定，下面就这 2 个阈值的选取讨论一下。

1) *LowestScore* 的选取：*LowestScore* 是用来过滤掉对于主题分类没有太大影响的噪声特征，本文

的实验过程如下：首先将样本集平均分为 5 份，然后每次选取其中 4 份作为训练集，另一份作为测试集，进行 5 次实验；在每次训练时，选取通过公式(5)得到的最大的区分度评分 $\max(DC(a_i))$与最小的区分度评分 $\min(DC(a_i))$，利用公式(10)计算出步长，然后从 $\min(DC(a_i))$+*step* 开始，每次增加 *step*，增加 4 次，算出每次的 *Macro-F* 值，在一共进行了 25 次实验后，最终的实验的结果见图 2。

$$step = \frac{\max(DC(a_i)) - \min(DC(a_i))}{6} \tag{10}$$

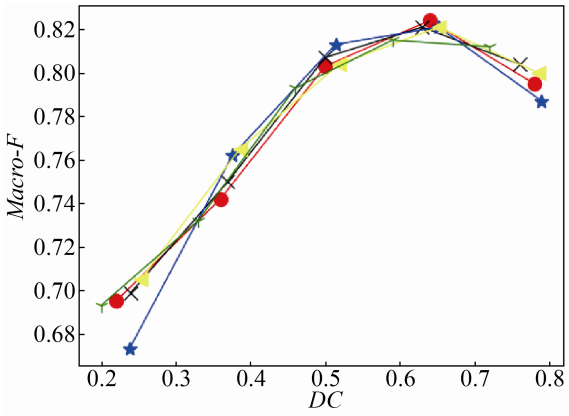

图 2 不同阈值效果图
Fig. 2 Different threshold effects

从图 2 发现，当 *LowestScore* 的值处于 0.6 到 0.7 之间时，分类的结果达到最高水平。通过上面的实验，我们还不能确定最优的阈值，因为每轮实验只选取了 5 个值，而 *LowestScore* 整体的区间约为(0.08, 0.93)，在利用公式(10)计算之后，落在(0.6, 0.7)区间内的值每次最多只有一个，所以我们只能确定一个最优的解空间。为了尽可能的逼近最优值，我们将实验的区间选为(0.6, 0.7)，以 0.1 为步长，进行 11 次实验，实验结果图 3 所示。从图 3 可以看出，当 *LowestScore* 取值为 0.63 时，实验的结果可以达到最优，因此本文将 *LowestScore* 值设定为 0.63。

2) *ScreeScore* 的选取：*ScreeScore* 是为每个特征筛选出相应的候选类别集合。当特征 a_i 对类别 c_k 的相对贡献率 Ca_{ik} 大于 *ScreeScore* 时，类别 c_k 就可以进入 a_i 的候选类别集。为了使得这个阈值对 Ca_{ik} 有一定的过滤功能，它的取值必须处于 $\max(Ca_{ik})$ 与 $\min(Ca_{ik})$ 之间，因此本文首先将 *ScreeScore* 确定为 $\overline{C_i}$ (Ca_{ik} 的均值)，并做了一些实验，发现效果还不错。

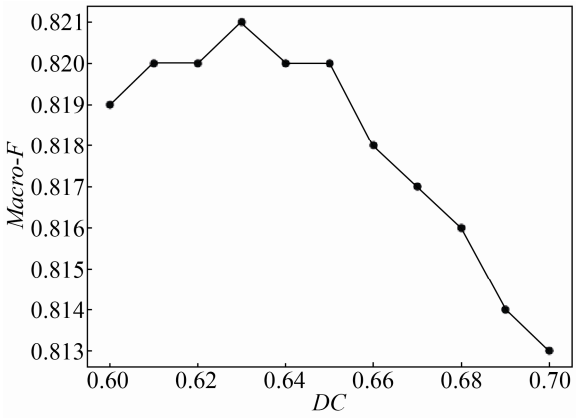

图 3 不同阈值效果图
Fig. 3 Different threshold effects

为了进一步确定最优的 *ScreeScore* 值，我们以 $\overline{C_i}$ 的 *n* 倍为自变量，*Macro-F* 为评价标准又进行了一系列的实验，结果如图 4 所示。

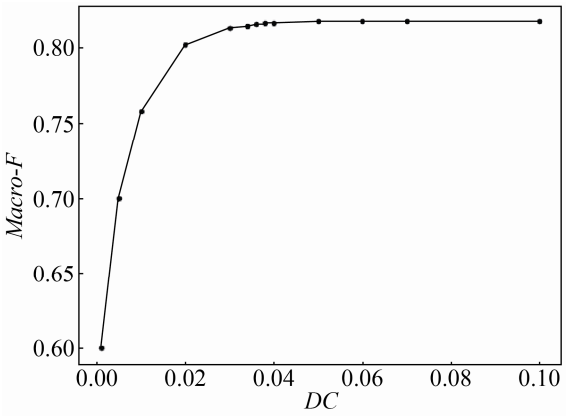

图 4 不同阈值效果图
Fig. 4 Different threshold effects

从图 4 我们发现：*n* 的值从 0.00~0.04 一直在以一个越来越弱的速率上升，而当 *n* 大于 0.04 之后，宏平均基本维持在一个很高的水平不再变化，而当 *n* 小于 0.02 时，实验的效果就不太理想了。本文的初始实验 *n* 的取值为 1(大于 0.04)，因此实验结果还算不错。综上，*ScreeScore* 值设定为 $\overline{C_i}$ 。

3.3 实验结果及分析

3.3.1 全网新闻数据集结果及分析

全网新闻数据共分为 20 个类目，每个类目有 2 000 篇文章，共 40 000 篇文章。文章长度大小不一，有的只有几十字，有的几千字，这样便于模拟网络数据，考察算法的扩展性。

本实验采用的是 5 次 5 折 1 交叉验证法，实验的最终结果取的是多次实验的均值，为了更加明显的展示算法的性能，选取了 CHI[15-16](chi-square)、IG[17] (Information Gain)、MI[18](Mutual information)、ECN[19] (Eliminate class noise)等算法作为对比。

图 5~6 显示的分别是 5 种算法在不同类别噪声比例下经朴素贝叶斯分类器与 SVM 分类器分类过的结果。所谓类别噪声比例就是类别噪声样本占全体样本的比例，当类别噪声比例为 0.5 时，表示样本中有一半的样本属于目标类别，当类别噪声比例为 0.8 时，表示样本中有 20%的样本属于目标类别。

ECN 算法与本文提出的 CNC 算法主要针对的是大规模层次分类，即当分类的主题越多时，分类的性能越优秀。这一点可以很明显的在图 5~6 上体现出来，当噪声类别的比例超过 70%时，ECN 与 CNC 相比于其他算法要更加优秀，当噪声比达到 90%以上时，这种优势更加明显。当类别噪声比例较低时(小于 0.6)，CNC、ECN、IG 三种算法的性能差不多，MI 与 CHI 则显得要逊色一点。如果只比较 CNC 与 ECN 算法，我们发现不管类别噪声比例大或小，CNC 算法的宏平均值始终要高一点。

在比较了准确率之后，本文也对分类器的响应速度进行了对比。因为 CHI、IG、MI 三种方法是传统的特征选择算法，它们并没有对类别噪声进行裁剪，导致每次分类决策调用了所有的分类器，使得分类速度很慢，所以速度实验就不用这 3 种算法进行对比，而是选取同样是类别噪声裁剪算法的 ECN 进行对比。对比采用的评价指标是分类器运行频率，即在分类相同大小的测试集情况下每个算法调用分类器的次数(分类器采用的是 SVM 分类器)。例如：在没有采用类别噪声裁剪算法的情况下，对 8 000 篇文档进行分类，需要调用 8 000×20 (主题数量)次分类器，而采用了剪枝算法之后，候选类别中的主题数量远远小于规定的主题数量，这样就加快了分类器的响应速度，而且文档与主题数量越大效果越明显。

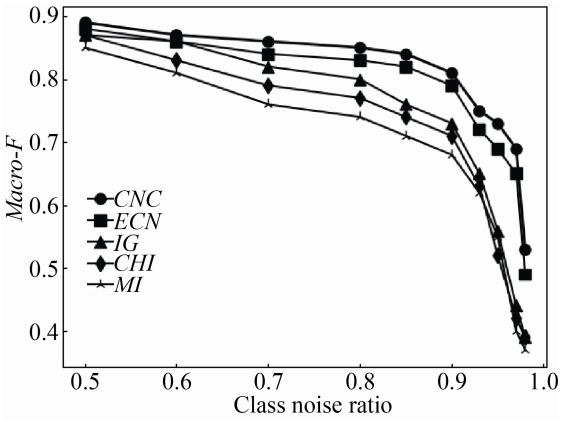

图 5 朴素贝叶斯分类器实验结果
Fig. 5 Experiment results of naive bayes classifier

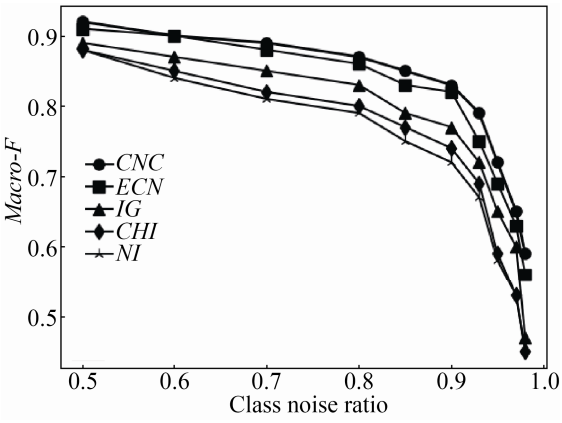

图 6 SVM 分类器实验结果
Fig. 6 Experiment results of SVM classifier

图 7~8 显示的是 CNC 与 ECN 算法在不同维度下，*Macro-F* 值与分类器运行频率的变化，选取的部分是每个算法 *Macro-F* 值最高的部分。本文在 3.2 节分析复杂度的时候提到过剪枝算法主要的时间复杂度集中在步骤(7)与(13)，而实验中横坐标所示的特征维度也是步骤(7)时间复杂度的一个方面，因为特征词典维度越大，文档中特征词被选中

的可能性越大，需要存储的东西越多。我们发现 ECN 算法性能最好的区域集中在维度 50 000 左右；而 CNC 算法性能最好的区域集中在维度 7 000 左右，这说明了在步骤(7)，CNC 算法的时间复杂度要优于 ECN 算法的优越性。

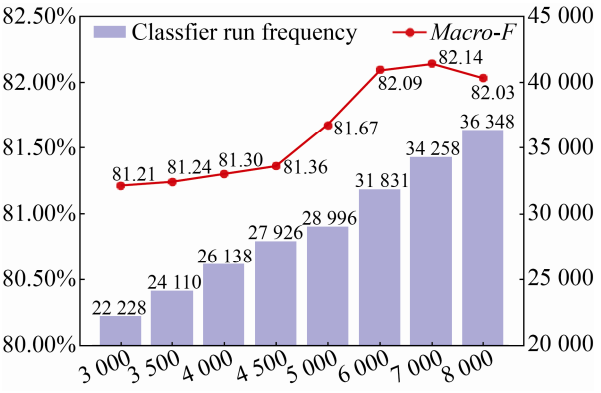

图 7 CNC 算法分类器使用频率与相应性能

Fig. 7 Frequency and corresponding performance of CNC algorithm classifier

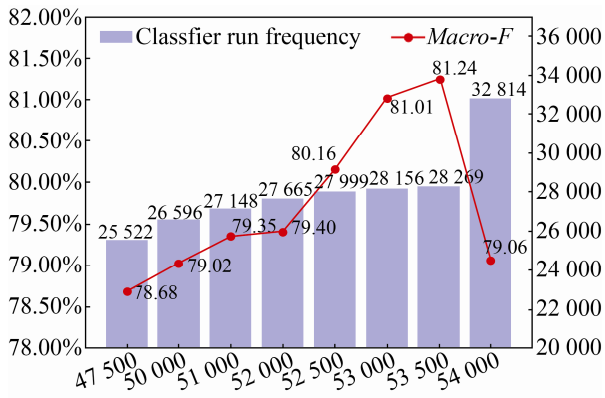

图 8 ECN 算法分类器使用频率与相应性能

Fig. 8 Frequency and corresponding performance of ECN algorithm classifier

再仔细观察图 7~8，发现在 *Macro-F* 值最高的部分，CNC 算法的分类器使用频率达到了 35 000 左右，而 ECN 算法的使用频率只有 28 000，这说明在性能峰值，ECN 算法在使用频率这一项上要优于 CNC 算法。但是，当 CNC 算法的特征维度取到 3 000~4 500 之间时，分类器的使用频率只有 25 000 左右，并且它的 *Macro-F* 值要高于 ECN 算法在所有维度上的性能最优值，这对应了步骤(13)的时间复杂度，说明在某个特征维度区间上 CNC 算法的准确率与响应速度可以同时超越 ECN 算法在全区间上的准确率与响应速度。

3.3.2 Reuters 数据集结果及分析

Reuters 一共有 21 578 条数据，经过 ModApte 划分之后，有许多记录由于 TOPICS 一栏中没有信息或者 BODY 一栏中没有信息，导致无法进行分类，只得丢弃，余下 15 673 条数据，取其中 12 000 条作为训练集，3 000 条作为测试集进行实验。

图 9~10 显示的是 5 种算法对 Reuters 数据集进行特征提取之后分别经过朴素贝叶斯与 SVM 分类器分类的结果。我们发现当噪声类别比例小于 0.5 时，ECN、CNC、IG 三种算法的分类效果要优于另外 2 种算法；而当噪声比例大于 0.6 时，CNC、ECN 两种层次分类算法的优越性渐渐展现出来；当噪声比例大于 0.9 时，这种优势达到了 10%，这些与中文数据集的实验结果特点相似。

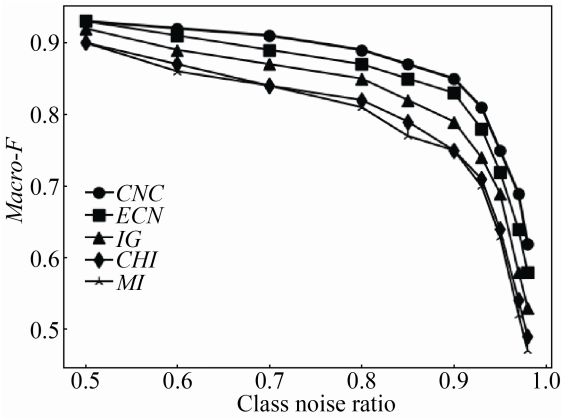

图 9 朴素贝叶斯分类器实验结果

Fig. 9 Experiment results of naive bayes classifier

图 11~12 显示的是 CNC 与 ECN 算法的分类器使用频率与总体性能，我们发现 CNC 算法的整体特征维度还是比 ECN 算法的要低很多，这样就节省了对每个文档遍历词典时的时间复杂度。*Macro-F* 值也是 CNC 算法的要高一些。另外，在 *Macro-F* 值最高的特征维度区间内，CNC 算法的分类器使用频率始终比 ECN 算法要低，而且体现出维度越高，频率次数少的越多的趋势。

DOI: 10.16182/j.issn1004731x.joss.19-FZ0289

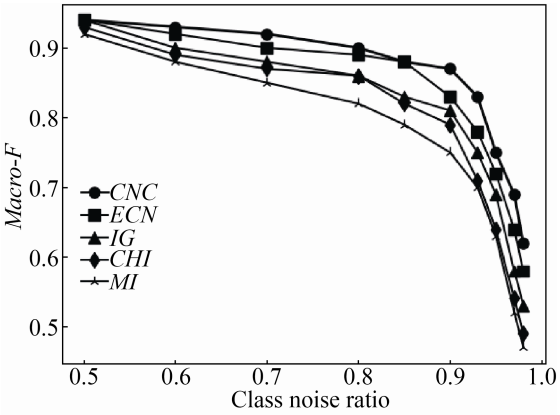

图 10 SVM 分类器实验结果
Fig. 10 Experiment results of SVM classifier

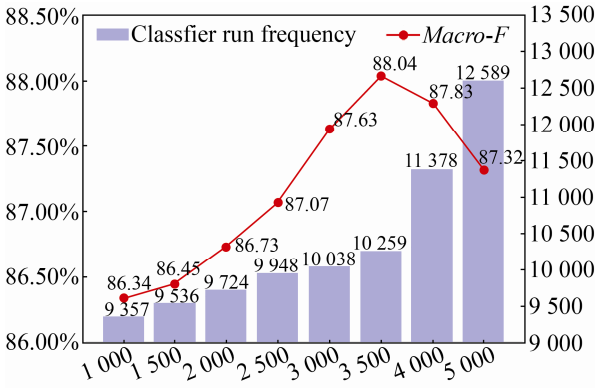

图 11 CNC 算法分类器使用频率与相应性能
Fig. 11 Frequency and corresponding performance of CNC algorithm classifier using

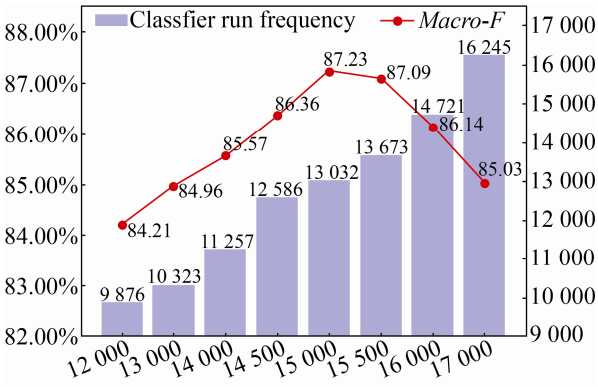

图 12 ECN 算法分类器使用频率与相应性能
Fig. 12 Frequency and corresponding performance of ECN algorithm classifier using

与中文数据集相比，我们发现巅峰性能的 CNC 算法比 ECN 算法的分类器使用频率要少得多，这可能是因为英文单词不像中文单词有那么多的歧义词，从而使得挑选出的特征词与少量的类别相关性较高，导致了最终每个特征词对应的候选类别数量相应减少。

4 结论

本文提出的基于相对贡献率的噪声裁剪算法将分类过程分为了搜索与分类两个阶段，首先利用搜索阶段挑选出对于分类价值较大的特征及其候选类别集，然后利用候选类别集对文档进行分类。实验证明这种方法与传统的算法相比在准确率与响应速度上都有较大的提升。为了凸显该算法的优越性，本文又与另一种噪声裁剪算法 ECN 进行了多方面的对比，发现不管是 *Macro-F* 值、分类器使用频率、巅峰性能特征维度数等方面都更加优秀。

本方法也有一定的缺点，例如剪枝过程中可能由于依据算法本身的局限性使得特征词相关的类别被排除在候选类别之外，这样就使得分类阶段候选类别中不存在测试文档的真实类别，因此，后续将主要研究如何设定一个回溯机制可以将候选类别进行重新筛选，进一步提高分类的效率。

参考文献：

[1] Du Y P, Liu J X, Ke W M, et al. Hierarchy Construction and Text Classfication Based on the Relaxation Strategy and Least Information Model[J]. Expert Systems with Applications (S0957-4174), 2018, 100: 157-164.

[2] 何力，贾焰，韩伟红. 大规模层次分类问题研究及其进展[J]. 计算机学报, 2012, 35(10): 2101-2115.
He Li, Jia Yan, Han Weihong. Category Candiadte Search in Large Scale Hierarchical Classification[J]. Chinese Journal of Computers, 2012, 35(10): 2101-2115.

[3] Sun A X, Lim E P. Hierarchical text classification and evaluation. Proceedings of the 2001 IEEE International Conference on Data Mining[C]. California, USA. 2001: 521-528.

[4] Liu T Y, Yang Y M, Wan H, et al. Support vector machines classification with a very large-scale taxonomy[J]. ACM SIGKDD Explorations Newsletter (S1931-0145), 2005, 7(2): 36-43.

[5] Silva-Palacios D, Ferri C. Improving Performance of Multiclass Classfications By Including Class Hierarchies [J]. Procedia Computer Science (S1877-0509), 2017, 108:

1692-1701.

[6] Luengo J, Shim S O, Alshomrani S, et al. CNC-NOS: Class Noise Cleaing By Ensemble Filtering and Noise Scoring[J]. Knowledge-Based Systems (S0950-7051), 2018, 140: 27-49.

[7] Zhuge H, He L. Automatic Maintenance of Category Hierarchy[J]. Future Generation Computer Systems (S0167-739X), 2017, 67: 1-12.

[8] Pereira R B, Plastino A, Zadrozny B, et al. Correlation Analysis of Performance Measure for Multi-label Classfication[J]. Information Processing & Management (S0306-4573), 2018, 54(3): 359-369.

[9] Kumar V, Pujari A K, Padmanabhan V, et al. Multi-label Classfication using Hierarchical Embedding[J]. Expert Systems with Applications (S0957-4174), 2018, 91: 263-269.

[10] Truică C O, Darmont J, Boicea A, et al. Benchmarking Top-k Keyword and Top-k Document Processing With T^2K^2 and $T^2K^2D^2$[J]. Future Generation Computer Systems (S0167-739X), 2018, 85: 60-75.

[11] Jiang L X, Wang S S, Li C Q, et al. Structure Extended Multinomial Naive Bayes[J]. Information Science (S0020-0255), 2016, 329: 346-356.

[12] Zhang L G, Jiang L X, Li C Qun, et al. Two Features Weighting Approachers For Naive Bayes Text Classfiers[J]. Knowledge-Based Systems (S0950-7051), 2016, 100: 137-144.

[13] Zhang D, Jiao L C, Bai X, et al. A Robust Semi-supervised SVM Via Ensemble Learning[J]. Applied Soft Computing (S1568-4946), 2018, 65: 632-643.

[14] Gaudioso M, Gorgone E, Labbé M, et al. Lagrangian Relaxation For SVM Feature Selection[J]. Computers & Operation Research (S0305-0548), 2017, 87: 137-145.

[15] Rempała G A, esołowski J. Double Asymptotics for the Chi-square Statistic[J]. Statistics & Probability Letters (S0167-7152), 2016, 119: 317-325.

[16] Ikram S T, Cherukuri A K. Intrusion Detection Model Using Fusion of Chi-square Feature Selection and Multi Class SVM[J]. Journal of King Saud University - Computer and Information Science (S1319-1578), 2017, 29(4): 462-472.

[17] Jadhav S, He H M, Jenkins K. Information Gain Directed Genetic Algorithm Wrapper Feature Selection For Credit Rating[J]. Applied Soft Computing (S1568-4946), 2018, 69: 541-553.

[18] Bennasar M, Hicks Y, Setchi R. Feature Selection Using Joint Mutual Information Maximisation[J]. Expert Systems with Applications (S0957-4174), 2015, 42(22): 8520-8532.

[19] 王强, 关毅, 王晓龙. 基于特征类别属性分析的文本分类器分类噪声裁减算法[J]. 自动化学报, 2007, 33(8): 809-816.
Wang Qiang, Guan Yi, Wang Xiaolong. A Method for Eliminating Class Noise In Text Classification Based on Feature Class Attribute[J]. Acta Automatica Sinica, 2007, 33(8): 809-816.

DOI: 10.16182/j.issn1004731x.joss.19-FZ0289