第41卷第7期 中国科学技术大学学报 Vol. 41, No. 7
2011年7月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Jul. 2011

文章编号:0253-2778(2011)07-0607-08

基于尾项加权的自适应文本分类方法研究

赖英旭,许 昕,杨 震

(北京工业大学计算机学院,北京 100124)

摘要:基于朴素贝叶斯分类框架,通过添加尾项值对部分严重扭曲的分类结果进行调整,达到提升分类器性能的目的.方法通过增量式自适应学习分类模式,根据历史结果,判断分类器分类质量,进而确定尾项添加区间,对明显产生分类扭曲的区间结果自适应添加尾项补偿,调整分类结果.在Trec05,Trec06,Trec07,Ceas08数据集上的对比实验表明,改进算法在accuracy,Macro F_1 两个指标上均比朴素贝叶斯分类器和bagging朴素贝叶斯分类器显著提高,且方法简单易行.

关键词:文本分类;朴素贝叶斯分类器;垃圾邮件过滤;尾项加权

中图分类号:TP391.1 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2011.07.007

Adaptive adjustment weighted text classification

LAI Yingxu, XU Xin, YANG Zhen

(*College of Computer Science and Technology, Beijing University of Technology, Beijing* 100124, *China*)

Abstract: To improve the performance of the naive Bayes classifier, a method is proposed which regulates text categories by adding adjustment values to the output of the naive Bayes classifier. The classification pattern was learned in an incremental and adaptive way, and the interval during which the output of the naive Bayes classifier should be adjusted was built according to the classification performance evaluated by historical outputs. Then the adjustment value was adaptively added to the output of the naive Bayes classifier distributed in the interval to regulate its category. The experiment results on Trec05, Trec06, Trec07, CEAS08 datasets show that the proposed method outperforms the naive Bayes classifier and the bagging naive Bayes classifier in terms of accuracy, Macro F_1, in addition to its simplicity and practicality.

Key words: text classification; naive Bayes; spam filtering; adaptive adjustment

0 引言

分类是数据挖掘与机器学习中一项非常重要的工作,在互联网数据处理中有着广泛的应用,如垃圾邮件过滤、新闻自动分类、博客倾向性判断等.构造分类器的方法有很多,常见的方法有贝叶斯方法、决策树方法、支持向量机等.其中,贝叶斯方法以其丰富的概率表达能力、综合先验知识的增量学习特性等成为众多方法中最为引人注目的焦点之一.鉴于学习最优贝叶斯分类器是一个NP-hard问题[1],而朴素贝叶斯分类器是一种构造简单的分类器,在很多情况下能够取得和一些相对复杂的分类器相当的分类性能,因此关于朴素贝叶斯分类器的研究工作得到广大学者的重视.但朴素贝叶斯分类器模型基

收稿日期:2011-04-28;**修回日期**:2011-06-21

基金项目:国家自然科学基金(61001178),北京市自然科学基金(4102012),北京市教育委员会科技发展计划面上项目(KM200810005030),北京市高等学校人才强教深化计划"中青年骨干人才培养计划"项目(PHR201108016),北京工业大学青年科学基金资助.

作者简介:赖英旭(通讯作者),女,1973年生,博士/副教授.研究方向:信息安全.E-mail:laiyingxu@bjut.edu.cn

于一个简单直观的假设:样本的各属性之间类条件独立.这影响了朴素贝叶斯分类器的性能.那么,通过放宽朴素贝叶斯分类器的属性间的类条件独立性,是否可以提高分类器的性能呢?为此,许多学者做了大量的研究工作并取得了一定成果.目前针对朴素贝叶斯分类器的改进方法主要集中在3个方面:①结构扩展;②特征选择;③属性加权.但是,采用结构扩展方法,除了需要计算属性的类条件概率,还需要计算属性之间的条件概率,增加了计算复杂度;采用特征选择方法,运用不同的标准进行特征选择得到的分类器性能也有所不同;采用属性加权方法,运用不同的方式计算得到的权值不同,对分类器性能的影响也不同.

本文分析了特征选择方法和属性加权方法.这两种方法通过公式变形后,在形式上可以等效成对朴素贝叶斯分类器的分类结果进行尾项加权.采用尾项加权的方法可避免大幅增加计算复杂性;也可避免因特征选取标准不同、属性权值的计算方法不同而造成的对分类器性能提升不同的问题.在此基础上,本文设计了基于尾项加权的朴素贝叶斯分类器,实验表明通过尾项加权可以提高朴素贝叶斯分类器的性能.

1 朴素贝叶斯分类器简介及相关改进工作

1.1 朴素贝叶斯分类器简介

假设 $A_1, A_2, \cdots, A_n$ 是数据集的 n 个属性,类型集合 C 包含有 m 个类 $C=\{c_1, c_2, \cdots, c_m\}$,给定一个待测样本 E,由向量 $\langle a_1, a_2, \cdots, a_n\rangle$ 表示,这里 a_i 为属性 A_i 的取值.$c(E)$ 表示 E 经分类后的类型标签.贝叶斯分类器定义如下:

$$c(E) = \arg\max_{c_t \in C} P(c_t) P(a_1, a_2, \cdots, a_n \mid c_t) \quad (1)$$

朴素贝叶斯假定样本属性之间类条件独立,定义如下:

$$c(E) = \arg\max_{c_t \in C} P(c_t) \prod_{i=1}^{n} P(a_i \mid c_t) \quad (2)$$

式中,$P(c_t)$ 称为先验概率,可以通过公式 $P(c_t)=\frac{D_t}{D}$ 计算,其中 D_t 表示训练集中类别为 c_t 的样本数,D 表示训练集中总的样本数.$P(a_i \mid c_t)$ 可以由训练样本估计得到,$P(a_i \mid c_t)=\frac{n_{it}}{n_t}$,其中 n_t 表示被标注为 c_t 类的样本所包含的所有属性的个数,n_{it} 表示在被标注为 c_t 类样本的属性集中属性 a_i 的个数.从式(2)可以看出,基于属性类条件独立假设,朴素贝叶斯的构造非常简单,但这也是朴素贝叶斯分类器的先天不足.为此科研工作者提出了许多改进算法来提升分类器性能.

1.2 朴素贝叶斯分类器的相关改进工作

目前针对朴素贝叶斯分类器的改进主要集中于以下3个方向:

(Ⅰ)结构扩展

该方法放宽属性的类条件独立假设,在构造分类器的过程中不但考虑了每个属性的类条件概率,同时也考虑了其他属性的影响,改变了朴素分类器模型的结构.经典算法如TAN算法[2].此外,蒋良孝等[3]于2005年提出ODANB(one dependence augmented naive Bayes)算法,每个属性 a_i 都与属性集合中另一个属性 a_j 存在关联,分类算法被定义为

$$c(E) = \arg\max_{c_t \in C} P(c_t) \prod_{i=1}^{n} P(a_i \mid a_j, c_t) \quad (3)$$

蒋良孝等[4]于2009年又提出HNB(hidden naive Bayes)算法,考虑属性集中所有其他属性对当前属性的影响,为属性节点构造一个隐性父节点,分类算法被定义为

$$c(E) = \arg\max_{c_t \in C} P(c_t) \prod_{i=1}^{n} \sum_{j=1 j\neq i}^{n} P(a_i \mid a_j, c_t) \quad (4)$$

(Ⅱ)特征选择

该方法通过去除属性集中冗余或者与分类无关的属性,从而达到提升分类器性能的目的.经典算法如SBC算法[5]、决策树算法[6].Meena等[7]在2009年提出在使用朴素贝叶斯分类算法对文本进行分类时,首先使用CHIR算法进行特征词选择,在选词过程中不仅计算了每个属性与类别之间的相关程度,还考虑了这种相关是正相关还是负相关;梁宏胜等[8]在研究中发现传统特征选择方法,如互信息、信息增益、期望交叉熵等,会产生不同的特征词集合,因此,他将互信息和期望交叉熵这两种方法相组合进行特征选择,实验证明该法能在一定程度上提高朴素贝叶斯分类器的分类性能.

(Ⅲ)属性加权

该方法不同于特征选择方法,后者将冗余或与分类无关的属性从属性集中完全剔除,而属性加权方法则是根据属性对分类的贡献为每个属性添加权值.Hall[9]于2006年提出了基于决策树计算属

性加权值的算法；Yager[10] 于 2006 年提出使用有序加权算子作为概率乘积的权重；蒋思伟等[11] 于 2008 年提出 WODANB(one dependence augmented naive Bayes)算法，该算法在 ODANB 的基础上引入属性加权，并利用 SVM 算法优化属性的权值.

2　基于尾项加权的朴素贝叶斯分类器

2.1　尾项加权方法原理简述

朴素贝叶斯假设样本各属性之间条件独立，研究者希望通过放宽特征之间的条件独立性来提高分类器的分类性能. 在不改变朴素贝叶斯模型结构的前提下，研究者更多地将研究重点集中在基于特征选择和基于属性加权这两类改进方法上.

基于特征选择朴素贝叶斯改进方法，是去除训练样本集中出现的冗余或者与分类不相干的特征，采用能为分类提供最大信息量的特征训练分类器. 该方法定义如下：

$$c(E) = \arg\max_{c\in C} P(c)\prod P(a_j \mid c) \tag{5}$$

式中，a_j 代表通过特征选择方法选出的特征.

与基于特征选择的改进方法不同，基于属性加权的改进方法没有将冗余的特征从训练特征集中除去，它采用的方式是在训练过程中计算出样本空间中每一个特征的权值，并在分类过程中考虑每个特征的权值. 该方法定义如下：

$$c(E) = \arg\max_{c\in C} P(c)\prod_{i=1}^{n} P(a_i \mid c)^{\omega_i} \tag{6}$$

式中，ω_i 表示特征 a_i 的权值.

对于二分类问题，可以采用对数似然比的形式来表示朴素贝叶斯决策规则[12]，其定义如下：

$$\mathrm{score}(E) = \sum_{i=1}^{n}\ln\left\{\frac{P(a_i \mid c_1)}{P(a_i \mid c_2)}\right\} + \ln\left\{\frac{P(c_1)}{P(c_2)}\right\} \propto \sum_{i=1}^{n}\ln\left\{\frac{P(a_i \mid c_1)}{P(a_i \mid c_2)}\right\} \tag{7}$$

式中，score(E)表示分类器对样本的评分. 决策规则为：若 score(E)＞0，则样本被分类成 c_1；若 score(E)≤0，则样本被分类成 c_2.

基于特征选择的朴素贝叶斯改进方法采用对数似然比的形式表示，形式如下：

$$\mathrm{score}'(E) = \sum_{i=1}^{n}\ln\left\{\frac{P(a_i \mid c_1)}{P(a_i \mid c_2)}\right\} - \sum_{t}\ln\left\{\frac{P(a_t \mid c_1)}{P(a_t \mid c_2)}\right\} + \ln\left\{\frac{P(c_1)}{P(c_2)}\right\} \tag{8}$$

式中，a_t 表示冗余的特征. 定义

$$\lambda = \ln\left\{\frac{P(c_1)}{P(c_2)}\right\} - \sum_{t}\ln\left\{\frac{P(a_t \mid c_1)}{P(a_t \mid c_2)}\right\} \tag{9}$$

则式(8)可以表示成如下形式：

$$\mathrm{score}'(E) = \sum_{i=1}^{n}\ln\left\{\frac{P(a_i \mid c_1)}{P(a_i \mid c_2)}\right\} + \lambda = \mathrm{score}(E) + \lambda \tag{10}$$

通过公式变换可以将基于属性加权的改进方法也变换成式(10)的表达形式.

经上述分析和公式推导，基于特征选择的朴素贝叶斯改进方法与基于属性加权的朴素贝叶斯改进方法在形式上可以等效成对朴素贝叶斯分类结果进行尾项加权. 其形式定义如下：

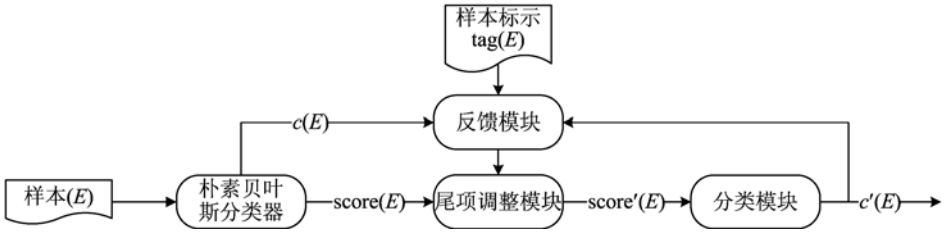
$$\mathrm{score}'(E) = \mathrm{score}(E) + \lambda \tag{11}$$

式中，score(E)表示朴素贝叶斯分类器给出的样本评分；score′(E)表示通过添加尾项加权后样本的评分；表示尾项权值.

2.2　基于尾项加权的朴素贝叶斯分类器算法

在上文提出的尾项加权方法基础上，本文设计了基于尾项加权的朴素贝叶斯分类器. 其工作流程如图 1 所示.

在训练朴素贝叶斯分类器时，如果采用的训练样本集中 c_2 类别的样本数量远远多于 c_1 类别的样本，则使用分类器对测试样本分类时，对于类别倾向不明显的测试样本，分类器倾向于将其类别判定成



E:样本，tag(E):样本 E 的标示，c(E):朴素贝叶斯对样本初分类的结果，c′(E):分类器最终输出的样本类别

图 1　基于尾项加权的朴素贝叶斯分类器流程图

Fig. 1　Naive Bayes classifier based on adjustment weighted

c_1. 对于这类样本，分类器输出的 score(E)大部分集中在分界线 0 附近的区域内，如图 2 所示.

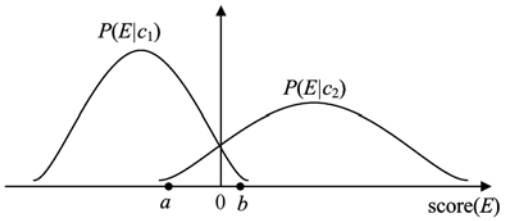


图 2 朴素贝叶斯分类器的分类错误域

Fig. 2 Error domain for the naive Bayes classifier

类别倾向不明显的样本的 score(E)集中于图 2 中所示的(a,b)区间内. 基于尾项加权的朴素贝叶斯分类器就是在朴素贝叶斯分类器输出的样本评分基础上，对落入区间(a,b)内的 score(E)添加尾项 λ，调整因分类器训练不均衡对样本分类造成的影响，从而达到提高朴素贝叶斯分类器分类性能的目的.

由于在增量式自适应学习过程中，朴素贝叶斯分类器的分类倾向性在经过了一段时间的波动后趋于稳定，图 2 所示的(a,b)也相应地经历了从波动到稳定的动态调整过程. 在自适应学习时，也有一些 score(E)落入(a,b)的样本被正确分类，对于落入区间内的样本进行尾项加权会出现以下 4 种情况：正确→错误，正确→正确，错误→正确，错误→错误. 因此需要反馈尾项加权后的分类结果，动态地调整尾项 λ 的值.

对尾项 λ 的值及尾项添加区间(a,b)的调整工作由反馈模块完成. 反馈模块分析 $c(E)$，tag(E)，$c'(E)$，并根据分析结果做出相应调整. 现在还需要确定对尾项调整的区间和范围，调整算法如算法 2.1 和算法 2.2 所示. 算法 2.2 所示尾项调整算法中，根据分类器的反馈信息、所处的不同状态以及所设置的风险因子 ξ 对尾项值 λ 采用不同的调整策略. 伪码中 fun(ξ)即为尾项调整策略函数，返回调整因子.

算法 2.1 尾项区间的调整算法

参数：score(E)朴素贝叶斯算法对样本的评分

$c(E)$朴素贝叶斯算法对样本的分类结果

tag(E)样本的真实类别

```
Algorithm RegionAdjust (score(E), c(E), tag(E))

begin
key=|score(E) / step|
numVec_class[key]←numVec_class[key]+1
If score(E)<=0
  class←c1
  the lower bound a will be changed
Else
  class←c2
  the upper bound b will be changed
If tag(E)=c(E)
the text is right classified
  return
Else
  the text is wrong classified
  wNumVec_class[key]←wNumVec_class[key]+1
  wNum←0
for i←0 to size of wNumVec_class do
  wNum←wNum+wNumVec_class[i]
  record the index i and the number wrong classifed wNum
  push the record back into intervalVec
  p←wNum / totalWrong_class
  If p >α
    tempKey←i
    break
for j←0 to tempKey
  bound←0
  numAll←numAll+wNumVec_class[j]
  for each element in intervalVec
    If element.key > j
      Break
    p←element.num / numAll
    If p >β
      bound←(element.key+1) * step
end
```

算法 2.2 尾项调整算法

参数：$c(E)$朴素贝叶斯算法对样本的分类结果

tag(E)样本的真实类别

$c'(E)$ 尾项加权调整后的分类

```
Algorithm WeightAdjust (c(E), tag(E), c'(E))

begin
  isStable=IsStable()
  If c(E)=tag(E) and c'(E)≠tag(E)
    If isStable=flase
      If len (λ) < len(domain)
        λ←fun1(ξ) * λ
      Else
        λ←fun2(ξ) * len(domain)
    Else
      If len (λ) < len(domain)
```

```
        λ←FuncStable(ξ)
      Else
        λ←tag(E) * len(domain)
  Else If c(E)≠tag(E) and c′(E)=tag(E)
    If isStable=flase
      λ←fun3(ξ) * λ
    Else
      λ←FuncStable(ξ)
      If λ has change flag
        λ←0
  Else
    If isStable=flase
      λ←fun4(ξ) * λ
    Else
      λ←λ+FuncStable(ξ)
      If len(λ) → len(domain)
        λ=(λ>0) ? len(domain):(-1) * len(domain)
end
```

3　实验结果与分析

3.1　实验数据与评测指标

实验采用的数据来自于文本检索会议(TREC)官方网站提供的 SPAM Track Trec05～07 语料库，以及 CEAS 2008 Live Spam Challenge 实验室语料库. 各语料库语料分布情况如表 1 所示.

实验采用增量式自适应学习方法训练分类器，即分类器使用现有规则分类样本，依据样本标示训练分类器更新分类规则库.

表 1　语料库组成

Tab. 1　Description of corpus used in the experiment

	Trec05	Trec06	Trec07	CEAS 08
总数	92 149	64 620	75 419	137 705
垃圾邮件数目	52 776	42 854	50 199	110 576
正常邮件数目	39 373	21 766	25 220	27 129

在文本分类实验中通常用精度(precision)、召回率(recall)、准确率(accuracy)3 个评测指标来评价分类器性能. 有时为更好地评价分类器针对某一类别样本的分类性能，需要综合考虑 precision，recall 这两个测试指标. 通常使用 F-measure[7]，定义如下：

$$F_\beta = \frac{(1+\beta^2)\times \text{recall}\times \text{precision}}{\beta^2\times \text{precision}+\text{recall}} \tag{12}$$

在大部分实验中，没有强调 precision，recall 哪一个指标更重要，所以 β^2 通常取为 1，记作 F_1. 若要评价分类器的整体分类性能需要对各个类别的 F_1 值求平均，得到宏平均 F_1. 定义如下：

$$\text{Macro } F_1 = \frac{F_{1_{C1}} + F_{1_{C2}}}{2} \tag{13}$$

本文采用 accuracy 和 Macro F_1 这两个指标作为分类器性能的评测指标.

3.2　实验结果

首先利用朴素贝叶斯分类器在 4 组语料库上进行分类实验，得到的实验结果如表 2 所示.

然后采用 Bagging 算法训练 9 个朴素贝叶斯分类器，每个朴素贝叶斯分类器的训练样本只占原训练样本的 80%. 对于每一个待测文本，首先由 9 个分类器进行分类，然后投票决定其类别. Bagging 算法在 4 组语料库上的实验结果如表 3 所示.

最后在同样的 4 组语料库上，使用基于尾项加权的朴素贝叶斯分类器进行分类实验，得到的结果如表 4 所示.

对比表 2、表 3、表 4 显示的实验结果，基于尾项加权的朴素贝叶斯分类器性能指标 accuracy，Macro F_1 的值最高，由此可见基于尾项加权方法可以提升朴素贝叶斯分类器的性能.

表 2　朴素贝叶斯分类器分类结果

Tab. 2　Experiment results of the naive Bayes classifier

corpus	category	precision	recall	accuracy	F_1	Macro F_1
Trec05	spam	0.991 47	0.925 023	0.952 501	0.957 095	0.951 951
	ham	0.989 333	0.907 784		0.946 806	
Trec06	spam	0.991 897	0.951 206	0.962 488	0.971 125	0.958 801 5
	ham	0.911 112	0.984 701		0.946 478	
Trec07	spam	0.997 197	0.963 864	0.974 144	0.980 247	0.971 417
	ham	0.932 563	0.994 608		0.962 587	
CEAS08	spam	0.999 955	0.799 685	0.839 12	0.888 677	0.799 359
	ham	0.550 482	0.999 853		0.710 041	

表 3 Bagging naive Bayes 分类器分类结果

Tab. 3 Experiment results of the bagging naive Bayes classifier

corpus	category	precision	recall	accuracy	F_1	Macro F_1
Trec05	spam	0.991 413	0.923 147	0.951 405	0.955 032	0.950 337
	ham	0.905 69	0.989 282		0.945 642	
Trec06	spam	0.991 152	0.948 896	0.960 491	0.967 681	0.955 697
	ham	0.907 172	0.983 322		0.943 713	
Trec07	spam	0.997 173	0.962 528	0.973 242	0.978 747	0.970 038
	ham	0.930 24	0.994 568		0.961 329	
CEAS08	spam	0.999 955	0.798 112	0.837 855	0.887 055	0.797 737
	ham	0.548 536	0.999 853		0.708 42	

表 4 加入尾项调整的朴素贝叶斯分类器分类结果

Tab. 4 Experiment result of the naive Bayes classifier based on adjustment weighted

corpus	category	precision	recall	accuracy	F_1	Macro F_1
Trec05	spam	0.977 298	0.943 762	0.955 236	0.960 237	0.954513
	ham	0.927933	0.970 614		0.948 793	
Trec06	spam	0. 991 468	0. 951 79	0.962 597	0.971 223	0.958902
	ham	0.912 014	0.983 873		0.946 582	
Trec07	spam	0.984 562	0.992 191	0.984 447	0.988 362	0.982463
	ham	0.984 213	0.969 033		0.976 564	
CEAS08	spam	0.998 316	0.943 378	0.953 255	0.970 07	0.931698
	ham	0.811 495	0.993 512		0.893 326	

比较这 3 种分类器的 accuracy 指标. 在 trec05, trec06, trec07 3 个语料库上, 基于尾项加权的改进算法对朴素贝叶斯分类器的性能提升效果没有在 CEAS08 语料库上体现得那么明显. 经过分析, 导致这种现象产生的原因有:

①计算 accuracy 时基数大. 性能指标 accuracy 的提升可以通过式(14)来计算:

$$\Delta\text{accuracy} = \frac{\Delta\text{num}_{\text{right}}}{\text{num}_{\text{all}}} \tag{14}$$

式中, $\Delta\text{num}_{\text{right}}$ 表示经尾项加权分类结果被修正的邮件数目; num_{all} 表示语料库中邮件数目. 从表 1 中可以看出每一个语料库所包含的邮件数量非常多, 即 num_{all} 很大. 此外, 朴素贝叶斯分类器在 trec05, trec06, trec07 3 个语料库上已经取得了很好的分类效果, 尾项加权方法只能对其中一部分邮件的分类结果进行调整, $\Delta\text{num}_{\text{right}}$ 相对较小. 所以在数值上 accuracy 指标提升不明显. 表 5 显示了通过尾项加权后修正的邮件数目.

表 5 通过尾项加权方法修正的邮件个数

Tab. 5 The number of emails revised by adaptive adjustment weighted method

	Trec05	Trec06	Trec07	CEAS08
修正的邮件数目	252	7	777	15 716

②采用的尾项添加区间及尾项值的调整算法对分类器性能的提升产生了一定的影响. 尾项加权方法是对落入尾项添加区间 (a,b) 的 score(E) 添加尾项 λ, 从而降低朴素贝叶斯分类器的分类倾向性对样本分类的影响. 首先需要确定尾项添加区间 (a,b). 定义

$$\text{ratio} = \frac{(a,b)\text{ 中被错分的邮件个数}}{(a,b)\text{ 中分类正确的邮件个数}} \tag{15}$$

本文采用对数似然比的形式来表示朴素贝叶斯分类: 当 score$(E)\in[0,+\infty)$, 则邮件被判定为 spam 类型; 当 score$(E)\in(-\infty,0)$, 邮件被判定为 ham 类型. 因此, 区间 (a,b) 的下界 $a<0$, 上界 $b\geqslant 0$. 确定尾项添加区间 (a,b) 的算法如下: 选择一个数值 L $(L<0)$ 作为界限, 在 $(L,0)$ 中确定一个数值 k, 若区域 $(k,0)$ 内 ratio 大于阈值 β, 则将 a 赋值为 k 作为区间的下界; 否则在 (L,k) 之间寻找新的数值, 直到找到下界或者不存在满足条件数值 k. 使用同样的方

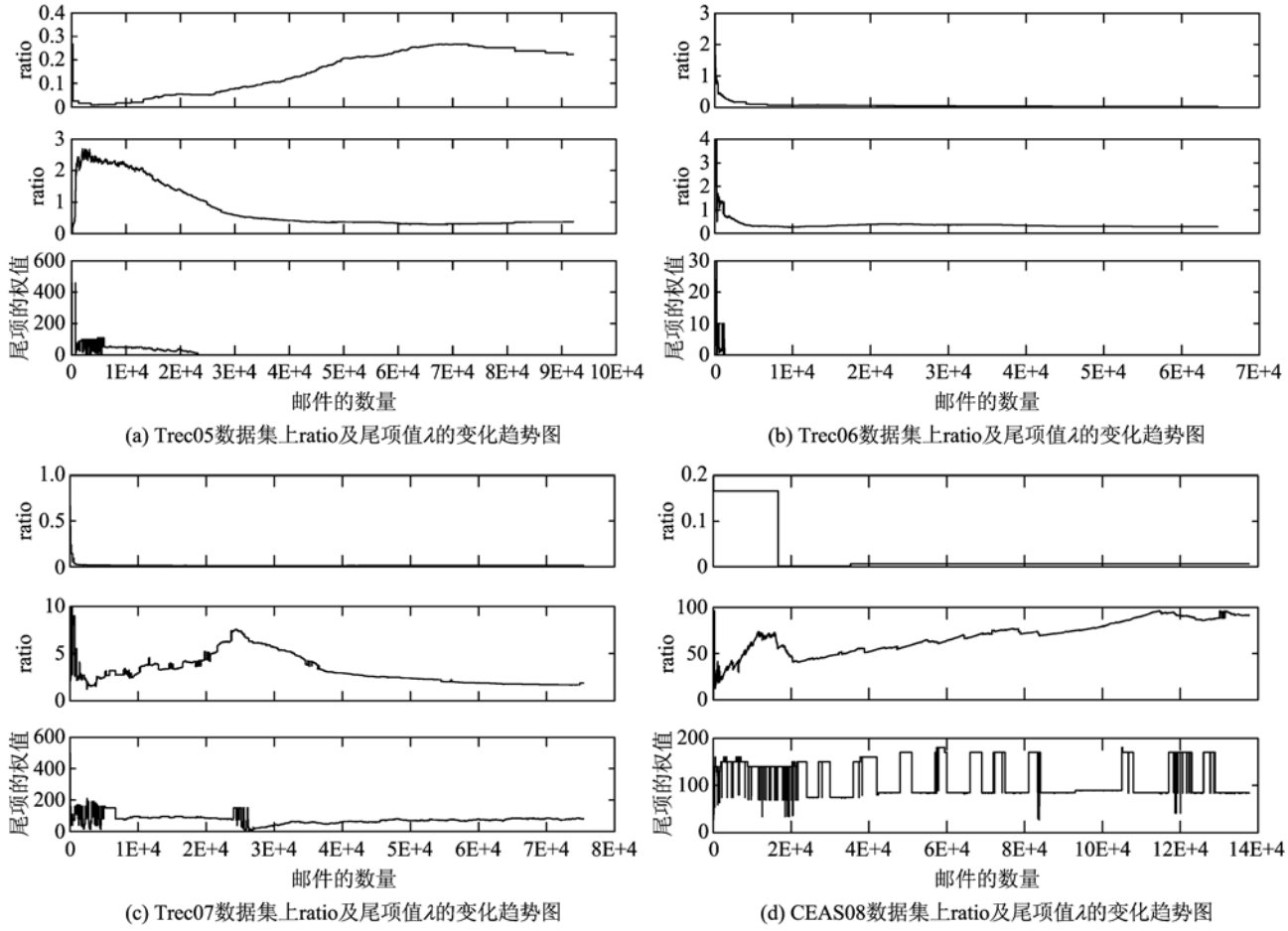


图 3　算法在不同数据集上测试性能分析图

Fig. 3　Performance of the algorithm on different corpus

法确定尾项添加区间的上界 b. 由于在增量式自适应学习过程中，ratio 的值不断地变化，区间 (a,b) 也处于动态调整中，直到 ratio 的值渐趋稳定，区间 (a,b) 才趋于稳定. 其次，需要确定尾项值 λ. 对落入区间 (a,b) 中的 score(E) 进行尾项加权，会将一部分原本分对的邮件分错，需要对尾项值 λ 进行调整. 当尾项添加区间稳定时，尾项值 λ 的调整范围才趋于稳定.

基于上述算法，若对于任意 k ($k<0$)，$(k,0)$ 中 ratio 都小于阈值 β，则下界 a 等于 0，即对于被分类成 spam 类型的邮件不进行尾项加权. 同理，若上界 b 等于 0，对于被分类成 ham 类型的邮件不进行尾项加权. 若上下界都为 0，则不再进行尾项加权，尾项值 λ 等于 0.

现设定阈值 $\beta=1$，设定寻找区间 $(a,0)$ 过程的界限为 $L(L<0)$，使得当前 $(L,0)$ 中被错分的邮件个数占当前所有被错分为 spam 类型邮件数的 75%；设定寻找区间 $[0,b)$ 过程的界限为 L'，使得当前 $(0,L')$ 中被错分的邮件个数占当前所有被错分为 ham 类型邮件数的 75%. 使用设定好参数的分类器在 Trec05，Trec06，Trec07，CEAS08 语料库上进行分类实验，绘制出分类过程中，区间 $[0,L')$ 中的 ratio、区间 $(L,0)$ 中的 ratio 以及尾项值 λ 的变化趋势图，如图 3 所示. 图 3 中第一个子图显示的是区间 $[0,L')$ 中 ratio 的变化趋势；第二个子图显示的是区间 $(L,0)$ 中 ratio 的变化趋势；第三个子图显示的是尾项值 λ 的变化趋势.

图 3(a)所示为分类器在 Trec06 语料库上的实验结果. 在区间 $[0,L')$ 中，ratio 的值均小于 1，说明当前在区间 $[0,L')$ 中，分类错误的邮件个数小于分类正确的邮件个数，则尾项添加区间 (a,b) 的上界 b 等于 0，分类器将不会对 score$(E)>0$ 的分类结果添加尾项. 在区间 $(L,0)$ 中，ratio 刚开始时大于 1，在经过一段波动后逐渐趋于 0. 在这段过程中，下界

a 起始值不为 0，即 $(a,0)$ 中分类错误的邮件个数大于分类正确的邮件个数，分类器对落入区间 $(a,0)$ 的分类结果 score(E)(score$(E)<0$)添加尾项. 当 ratio 小于 1 后，下界 a 为 0，分类器将不再对分类结果添加尾项，尾项值为 0. 由于尾项添加区间 (a,b) 经过很短时间的调整后就变为(0,0)，说明分类器发生 ham→spam，spam→ham 这两类错误的概率大致相同，所以分类器只对很少的一部分分类结果通过尾项加权进行调整. 因此尾项加权方法在 Trec06 语料库上对于朴素贝叶斯分类器的性能改进较小.

分类器在 CEAS08 语料库上的实验结果如图 3(d)所示. 在区间 $[0,L')$ 中，ratio 的值均小于 1，说明当前在区间 $[0,L')$ 中分类错误的邮件个数远小于分类正确的邮件个数，尾项添加区间 (a,b) 的上界 b 等于 0，分类器将不会对 score$(E)>0$ 的分类结果添加尾项. 在区间 $(L,0)$ 中，ratio 的值远大于 1 且呈现上升趋势，则尾项添加区间的下界 a 在试验过程中不为 0，说明在区间 $(a,0)$ 中分类错误的邮件个数远大于分类正确的邮件，即分类器发生 spam→ham 这类错误的概率很大. 对落入区间 $(a,0)$ 中的分类结果进行尾项调整，可以将大量原本被错分为 ham 类型的邮件重新正确分类为 spam 类型，从而大大提高了分类器的性能.

4 结论

尾项加权方法是对朴素贝叶斯分类器的输出结果进行尾项加权，调整分类器因训练不均衡而产生的分类倾向性对样本分类的影响. 运用此方法，本文设计了基于尾项加权的朴素贝叶斯分类器，并与朴素贝叶斯分类器在 trec05，trec06，trec07，CEAS08 语料库上进行对比实验. 通过对实验结果的观察与分析发现，当朴素贝叶斯分类器对于样本分类存在较为严重的分类倾向性时，尾项加权方法可以很好地调整分类器的这种分类倾向性，而且简单易行.

通过尾项加权可以调整朴素贝叶斯分类器的分类倾向，在此基础上，可以通过设定尾项值以及相应的尾项调整策略来定制分类器，以满足特定的需求. 例如，将朴素贝叶斯分类器应用于垃圾邮件过滤时，要求在保证整体分类性能不下降的前提下，减少正常邮件被错判为垃圾邮件的数目. 今后的工作将在尾项加权方法的框架下，研究影响分类器性能的因素与尾项之间的关系，制定更加通用的尾项设定和调整方案，以求通过简单操作就能定制满足需求的分类器.

参考文献(References)

[1] Chickering D M. Learning Bayesian networks is NP-complete [M]// Learning for Data: Artificial Intelligence and Statistics V. New York: Springer-Verlag, 1996:121-130.

[2] Friedman N, Geiger D, Goldszmid M. Bayesian network classifiers[J]. Machine Learning, 1997, 29: 131-163.

[3] Jiang L X, Zhang H, Cai Z H, et al. One dependence augmented naive bayes [J]. Lecture Notes in Computer Science, 2005, 3 584: 186-194.

[4] Jiang L X, Zhang H, Cai Z H. A novel Bayes model: Hidden naive Bayes [J]. IEEE Transactions On Knowledge and Data Engineering, 2009, 21(10): 1 361-1 371.

[5] Langley P, Sage S. Induction of selective Bayesian classifiers [C]// Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence. Seattle, WA, USA: Morgan Kaufmann Publishers, 1994: 339-406.

[6] Ratanamahatana C, Gunopulos D. Feature Selection for the Naive Bayesian Classifier Using Decision Trees [J]. Applied Artificial Intelligence, 2003, 17(5/6): 475-487.

[7] Meena M J, Chandran K R. Naive Bayes Text Classification with Positive Features Selected by Statistical Method [C]// Proceedings of the 1st International Conference on Advanced Computing. Chennai, India: IEEE Press, 2009: 28-33.

[8] Liang Hongsheng, Xu Jianmin, Cheng Yuepeng. An Improving Text Categorization Method of Naive Bayes [J]. Journal of Hebei University (Natural Science Edition), 2007, 27(3): 327-331.
梁宏胜，徐建民，成岳鹏. 一种改进的朴素贝叶斯文本分类方法[J]. 河北大学学报(自然科学版)，2007，27(3)：327-331.

[9] Hall M. A decision tree-based attribute weighting filter for naive Bayes[J]. Knowledge-Based Systems, 2006, 20(2): 120-126.

[10] Yager R R. An extension of the naive Bayesian classifier[J]. Information Sciences, 2006, 176(5): 577-588.

[11] Jiang S, Cai Z. Improve the accuracy of one dependence augmented naive Bayes by weighted attribute [J]. Lecture Notes in Computer Science, 2008, 5 370: 556-561.

[12] 边肇祺，张学工. 模式识别[M]. 2 版. 北京：清华大学出版社，2000：22-23.