

基于互信息规则剪枝的关联文本分类

商炳章, 白清源

(福州大学 数学与计算机科学学院, 福建 福州 350002)

[摘要] 传统的关联文本分类算法产生的规则数量巨大, 若不对规则剪枝会影响分类效率, 而采用以前的剪枝方法又会使分类精度出现不同程度的下降. 为此提出以互信息的方法对每个类的规则进行剪枝, 挑选出分类能力强的规则构成分类器, 对待分类文本进行分类. 经过这个方法剪枝后的规则数量大幅减少, 且能取得比规则集未修剪过的分类器和采用以前剪枝方法的 ARC-BC 算法更好的分类效果, 大量的实验表明此方法是有效的.

[关键词] 互信息, 规则剪枝, 关联分类

[中图分类号] TP18 [文献标识码] A [文章编号] 1672-1292(2008)04-0173-05

On Classification of Associative Text Based on Rules Pruning of Mutual Information

Shang Bingzhang Bai Qingyuan

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002, China)

Abstract The traditional associative classifying algorithms of associative texts generate a huge number of rules. If the rules were not pruned, the efficiency of classification would be influenced. However, if the former pruning method were adopted, different degrees of accuracy of classification would appear. Therefore, an associative text classification algorithm-based on rules pruning of mutual information is presented to prune the rules of each class. The rules with high classifying capacity are chosen to form classifiers to classify the texts being classified. The study illuminates that the mutual information-based rules pruning algorithm not only gets much less rules but is more helpful for improving the accuracy of the association categorization. The experimental results show the performance of this method is better than both ARC-BC algorithm and the algorithm which uses all rules.

Key words mutual information, rules pruning, associative classification

1998年, Liu Hsu和Ma提出关联分类方法CBA^[1]. CBA集成分类规则挖掘过程和关联规则挖掘过程, 取得比同样基于规则的决策树分类算法C4.5更好的分类效果. 此后, 针对CBA的不足提出各种变种, 典型的有CMAR^[2]和ARC-BC^[3]. 不像决策树等经典分类方法那样只产生少量的规则, 关联分类器在一定程度上会产生大量的规则集. 如果不对规则集进行剪枝, 大量的规则既浪费资源, 又影响效率. 当前大部分关联分类器通过规则剪枝技术对冗余规则进行修剪, 但修剪后的分类精度出现不同程度的下降, 下降的幅度与使用的修剪策略有关^[2,3]. 为此本文提出了以互信息的方法来对每个类挖掘出来的规则进行重新排序, 然后按照一定比例从每个类排序好的规则集中选取前 K 条分辨力强的规则构成分类器, 再对文本进行分类, 文本最后分到类置信度最高的类别中.

1 基本定义

定义 1 项: 文本的内容特征常用它所含有的特征词来表示, 这些词被称为项 (term), 表示成 t_i . 如果文本含有该特征词, 那么 $t_i = 1$; 否则 $t_i = 0$. 一个或一个以上的项组成项集.

收稿日期: 2008-06-18

基金项目: 教育部留学回国人员启动基金、中科院软件所开放课题基金 (SYSKF0701)、福州大学科技发展基金 (2005-XQ-13) 和福建省教育厅基金 (JB06023) 资助项目.

通讯联系人: 白清源, 教授, 研究方向: 数据库技术和数据挖掘. E-mail: baiqy@fzu.edu.cn

定义 2 文本的向量表示: 给定一个文本 $d = (t_1, t_2, \dots, t_n)$, 当暂时不考虑 $t_k (1 \leq k \leq n)$ 在文本中的先后顺序并要求 t_k 互异时, 可以把 t_1, t_2, \dots, t_n 看成是一个 n 维坐标系, 称向量 (t_1, t_2, \dots, t_n) 为文本 d 的向量表示, 其中 n 为所选取的特征词总数.

定义 3 规则支持数: 指训练集 D 中包含有项集 A 并且具有相同类标号 C_i 的文本个数.

定义 4 规则支持度 Support 指 D 中包含有项集 A 并且具有相同类标号 C_i 的文本个数占 C_i 类文本总数的百分比.

定义 5 规则置信度 Confidence 指 D 中包含项集 A 的文本被标记为类标号 C_i 的百分比.

2 分类关联规则的挖掘及剪枝

现有的基于关联规则的分类方法的基本思想是利用现有的关联规则挖掘算法 Apriori^[4] 或 FP-Tree^[5] 产生局部类别中的频繁项, 即频繁出现的特征词或特征词项集, 然后再用这些频繁项集为规则前件, 以类别名为后件构造分类规则, 并以此规则集构成分类器对测试样本进行分类: 如果测试样本与某类别的规则前件相匹配, 则把该规则的置信度累加到该类的计数器, 若某类别计数器的置信度之和最大, 则判定测试样本属于该类别.

2.1 关联规则的挖掘

在各种关联规则挖掘算法中, 最经典、最广泛使用的是 Apriori^[4] 和 FP-Growth^[5] 算法. Apriori^[4] 算法在关联规则挖掘过程中将每个文档当成 1 个事务, 把每个特征项当成事务项目, 然后从这些事务中挖掘出支持度大于设定的最小支持度 minsup 的频繁项集. 例如, 如果 {世锦赛, 篮球} 是“体育”这个类中的 1 个频繁模式, 其中世锦赛、篮球为特征词, 那么“{世锦赛, 篮球} \rightarrow 体育”为 1 条候选规则, 这条规则的前件长度为 2 下一步把所有类的规则融合在一起, 进行适当的剪枝后形成最后的规则集, 也就是关联分类器.

2.2 传统的规则剪枝方法

由于用 Apriori^[4] 算法挖掘的频繁模式的规模可能相当大, 所以需要对挖掘得到的关联规则进行剪枝, ARC-BC^[2] 的剪枝策略是只保留规则前件更一般且规则置信度更高的规则, 并利用数据库覆盖方法剪掉不能覆盖至少 1 个事务的规则. 例如:

$R_1: t_1, t_2 \rightarrow C_1$ [confidence 40%], $R_2: t_1 \rightarrow C_1$ [confidence 50%]. R_2 就是那些需要保留下来的规则, 而 R_1 则要被剪枝. 经过剪枝, 则剪去了那些对分类没有帮助的, 冗余的分类规则.

3 基于互信息的规则剪枝

3.1 互信息简介

互信息在信息论中是作为一种衡量 2 个信号关联程度的尺度, 后来引申为 2 个随机变量间的关联程度进行统计描述, 可表示成这 2 个随机变量的概率的函数. 假设 $MI(X, Y)$ 为随机变量 X 和 Y 的互信息, 则:

$$MI(x, y) = \log \frac{p(x, y)}{p(x)p(y)}, \text{ 其中: } p(x, y) = \frac{c(x, y)}{\sum_{x, y} c(x, y)}, p(x) = \frac{c(x)}{\sum_x c(x)}, p(y) \text{ 和 } P(y) \text{ 分别是 } x \text{ 和 } y \text{ 独}$$

立出现的概率, $P(x, y)$ 是 x 和 y 同时出现的概率. 当 $MI(x, y) \gg 0$ 时, 表明 x 和 y 的关联程度强; $MI(x, y) = 0$ 时, 表明 x 和 y 的关联程度弱, 他们的同时出现仅属于偶然; 当 $MI(x, y) \ll 0$ 时, 表明 x 和 y 互补分布, 不存在关联关系. $C(x, y)$ 表示 x 和 y 同现在 1 篇文档中的次数, 同理, $C(x)$, $C(y)$ 表示 x 和 y 在文档中出现的次数. 在关联文本分类中, 互信息已被广泛应用于特征词提取, 然而据作者所知, 目前还没有文献把互信息用于对挖掘出的规则进行剪枝. 本文对每个类挖掘出的规则计算该条规则和该类文档的互信息然后按照互信息值对每个类的规则进行排序, 并从每个类排序好的规则中选取前 k 条分辨能力强的规则组合起来以构成分类器, 实验结果显示该方法是有效的, 能够取得比 ARC-BC 分类器和未剪枝的分类器更好的效果. 当前大部分的关联分类器像 CBA, CMAR 和 ARC-BC 通过规则剪枝技术对冗余的规则进行修剪, 但修剪后的分类精度出现不同程度的下降^[2, 3], 令人兴奋的是本文所采用的基于互信息的剪枝方法不但没有使得分类精度下降, 反而在规则集变小的情况下取得更好的分类效果. 下面介绍本文所采用的规则和

某类文档的互信息的基本定义.

定义 6 C_i 类文档的出现概率: 设 D 是所有的文本向量的集合, C 为类标签属性, $D / IND(C) = \{C_1, C_2, \dots, C_m\}$, 其中 $D / IND(C)$ 表示为所有文本向量集合根据类标签属性值所得到的等价类划分, 则某文档类 C_i 的出现概率为:

$$p(c_i) = \frac{|C_i|}{|D|}. \quad (\text{公式 1}), \text{ 其中, } |C_i| \text{ 表示第 } i \text{ 类文档的数量, } |D| \text{ 表示所有文档的总数.}$$

定义 7 规则前件的出现概率: 指 D 中包含有项集 A 的文本个数占文本总数的百分比. 记为:
 $P(A) = \frac{|A|}{|D|}$ (公式 2), 其中 $|A|$ 表示包含有项集 A 的文本个数, $|D|$ 表示所有文档的总数.
 定义 8 规则和某文档类的互信息: 设规则 $R: t_1 \ t_2 \dots t_n \rightarrow C_k$, 规则的前件为 $T(t_1 \ t_2 \dots t_n)$, 规则的后件为 C_k . 则规则 R_i 和 C_k 类文档的互信息可以表示为:

$$MI(T, C_k) = \log \frac{P(T, C_k)}{P(T)P(C_k)}. \quad (\text{公式 3}), \text{ 其中 } P(T, C_k) \text{ 表示包含前件 } T \text{ 的 } C_k \text{ 类文档出现的概率即包含规则 } R_i \text{ 的文档个数占文本总数的百分比. 记为 } P(T, C_k) = \frac{|R_i|}{|D|}, \text{ 其中 } |R_i| \text{ 表示包含规则 } R_i \text{ 的文本个数, } |D| \text{ 表示所有文本总数. } P(T) \text{ 为规则前件 } T \text{ 的出现概率, 可由公式 2 计算得到. } P(C_k) \text{ 为 } C_k \text{ 类文档出现的概率, 可由公式 1 计算得到.}$$

下面, 举个例子来简要说明. 根据下面表 1 的数据, $D / IND(C) = \{\{1 \ 3 \ 5\}, \{2 \ 4\}\}$, 其中 $\{1 \ 3 \ 5\}$ 为 C_1 类的文本 Id 号集合, $\{2 \ 4\}$ 为 C_2 类的文本 Id 号集合. 假设 C_1 类文档的规则 R_1 为: $E, F \rightarrow C_1$, 那么根据表 1 的数据和前面的公式我们可以得到: C_1 类文档出现的概率为 $P(C_1) = \frac{|C_1|}{|D|} = \frac{3}{5}$, 有项集 $T(E, F)$

出现的概率为 $P(T) = \frac{|T|}{|D|} = \frac{2}{5}$, $P(T, C_1) = \frac{|R_1|}{|D|} = \frac{1}{5}$, 所以可以得到规则 R_1 和 C_1 类文档的互信息为:

$$MI(T, C_1) = \log \frac{P(T, C_1)}{P(T)P(C_1)} = \frac{\frac{1}{5}}{\frac{2}{5} \times \frac{3}{5}} = 0.8333$$

与计算两个变量之间的互信息不同, 本文在计算规则与类文档的互信息时把规则前件所包含的特征项集当作一个整体.

4 分类器的构造和分类测试文本

定义 9 类置信度^[6]: 定义与文本 d 匹配的所有指向类别 $C_j (1 \leq j \leq m)$ 的分类规则置信度之和为类别 C_j 的类

置信度, 即 $\Omega(d, C_j) = \sum_{i=1}^l \text{Confidence}(R_i \rightarrow C_j)$ ① 其中 l 为匹配的规则数.

我们称本文提出的文本分类算法为 Pruned by Mutual Information(简称 PMI), 下面是整个算法的步骤:
 算法说明: PMI 在 A priori 算法产生的规则集的基础上计算每条规则和规则所在文档类的互信息, 并按照一定比例选择每类规则中互信息值最大的前 K 条规则组合起来构成关联文本分类器对待分类文本进行分类, 以减少规则的数量并达到更好的分类效果.

算法输入: A priori 算法产生的规则集 Rule_Set 测试样本集合 Doc_Test

算法输出: 待分类文本所属的类别;

算法过程:

- (1) foreach Rule $R(T \rightarrow C_i)$ in Rule_Set do{
- (2) CountMutualInformation(R, C_i);
- (3) }
- (4) foreach Each type of rules RuleSet(C_i) do{

表 1 C_1, C_2 类所对应的数据库向量表

Table 1 The database vector of C_1, C_2 class

Id	A	B	C	D	E	F	G	Class/ C
1	0	1	0	1	1	1	0	C_1
2	1	0	1	0	1	1	0	C_2
3	1	0	1	1	1	0	0	C_1
4	1	1	1	0	1	0	1	C_2
5	0	0	0	1	1	0	0	C_1

```
(5)      按照一定的比例  $\beta$  选择互信息值最大的前  $K$  条规则构成分类器;
(6) }
(7)      foreach document  $D_i$  in Doc_Test do {
(8)      foreach Rule  $R(T \rightarrow C_i)$  in Rule_Set do {
(9)      If  $D_i$  与规则  $R(T \rightarrow C_i)$  的前件相匹配 then
(10)      把规则  $R$  的置信度累加到  $C_i$  类的置信度计数器中;
(11) }
(12)      把各个类的置信度计数器按照从大到小的顺序进行排序;
(13)      返回置信度之和最大的那个类  $C_i$ ;
(14) }
```

PM I 算法的改进主要是第 2 行和第 5 行. 第 2 行 $\text{CountMutualInformation}(R, C_i)$ 根据公式 (3) 计算每条规则和所在文档类的互信息. 第 5 行选择每个类别中互信息值最大的前 K 条规则构成分类器. 经过互信息剪枝后的规则集中的规则数量明显的减少, 却能取得更好的分类效果.

5 实验结果与分析

5.1 样本集的处理

我们在 P4 2.0, 768M 内存计算机上使用 VC++ 实现 PM I 算法, 实验数据是从中文自然语言处理开放平台网站获取李荣陆^[8]收集的新华社的新闻样本. 一共 2815 个样本, 10 个类别, 分别为环境, 体育, 教育, 交通, 计算机, 经济, 军事, 医药, 艺术, 政治; 利用 χ^2 Statistics 和基于文档统计取得文本特征, 特征维数设为 50 维. 从中选取 1881 个样本作为训练样本, 934 样本作为测试样本.

5.2 实验度量标准

使用目前常用的度量标准 Precision (P) 和 Recall (R), $F1^{[7]}$ 以及 Macro-avg (宏平均)^[7] 和 Micro-avg (微平均)^[7].

5.3 PM I 算法和其它分类器的比较

限于篇幅, 这里仅列出 β 取表中 4 个值的情况. 从表 1 可以看出, PM I 算法在开放测试中的准确率平均要比未经剪枝的分类器高 1.2% 左右, 比采用其它剪枝方法的 ARC-BC 算法高 2.6% 左右. 而封闭测试的准确率分别高出 1.2% 和 4.1%.

表 2 β 取值对分类准确率 P 的影响
Table 2 The influence of the value of β on the accuracy of classification P

	ARC-BC	AllRules(未剪枝)	PM I ($\beta=92\%$)	PM I ($\beta=89\%$)	PM I ($\beta=86\%$)	PM I ($\beta=83\%$)
P (开放测试)	53.2%	54.4%	54.9%	55.7%	56.1%	56.3%
P (封闭测试)	59.5%	62.4%	62.9%	63.6%	63.9%	64%

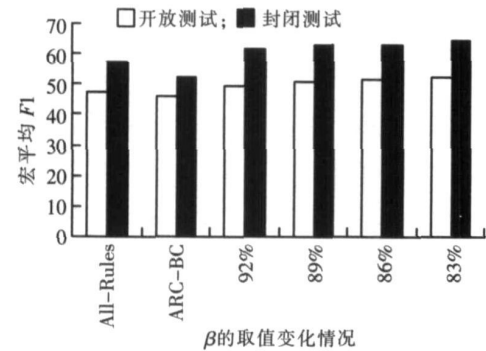


图 1 PMI 的宏平均 $F1$ 值随规则选取比例变化的比较
Fig.1 The transformation of macro- $F1$ with the change of the proportion of the rules

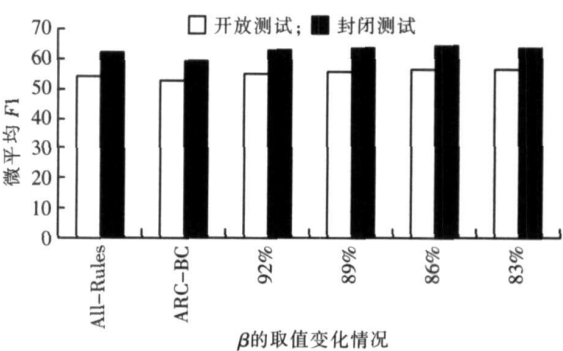


图 2 PMI 的微平均 $F1$ 值随规则选取比例变化的比较
Fig.2 The transformation of micro- $F1$ with the change of the proportion of the rules

从图 1 可以看出, PM I 算法在开放测试中的宏平均 $F1$ 平均要比未经剪枝的分类器高 3.5% 左右, 比采用其他剪枝方法的 ARC-BC 算法高 5.2% 左右; 在封闭测试中的宏平均 $F1$ 平均要比未经剪枝的分类

器高 5.4% 左右, 比采用其它剪枝方法的 ARC-BC 算法高 1.1% 左右. 从图 2 可以看出, PM I 算法在开放测试中的微平均 $F1$ 要比未经剪枝的分类器平均高 2% 左右; 比采用其它剪枝方法的 ARC-BC 算法平均高 2.3% 左右; 在封闭测试中的微平均 $F1$ 要比未经剪枝的分类器平均高 1.5% 左右; 比采用其它剪枝方法的 ARC-BC 算法平均高 4.5% 左右. 从图 3 可以看出 PM I 分类器所拥有的规则数比 All Rules ARC-BC 分类器的规则数少得多.

综上所述可以看出, PM I 算法能够在规则数量更少的情况下取得比未经剪枝和采用其它方法剪枝的分类器更好的分类效果. PM I 分类器取得比较好的结果, 可能的原因有: 1 All Rules ARC-BC 分类器中含有相同规则前件的规则数量较多, 导致其分辨能力较 PM I 分类器弱; 2 All Rules ARC-BC 分类器采用某些置信度高但是规则前件太长的规则来进行分类, 产生过度拟合的情况, 致使分类精度下降.

6 结论

针对关联分类器会产生大量的规则, 且效率比较低的问题, 本文提出了以互信息的方法来对每个类挖掘出来的规则进行重新排序, 然后从每个类排序好的规则集中按一定的比例选取前 K 条分辨力强的规则构成分类器对文本进行分类, 能够在规则集变小的情况下, 取得比未修剪过的规则集更好的分类精度, 同时也取得比 ARC-BC 分类器更好的分类效果. 大量的实验结果表明本文提出的方法具有较强竞争力. 与本文相关的几个开放问题还有待解决, 其中包括如何更加科学地确定每个类别规则数的选取值 K 等. 下一步将对这些问题进行研究.

[参考文献] (References)

- [1] Liu B, Hsu W, Ma Y M. Integrating classification and association rule mining [C] // ACM Int'l Conf on Knowledge Discovery and Data Mining. New York: ACM Press, 1998: 80-86.
- [2] Li W, Han J, Pei J. CMAR: Accurate and efficient classification based on multiple classification rules [C] // Cercone N. Proc of the 2001 IEEE Int'l Conf on Data Mining. California: IEEE Press, 2001: 369-376.
- [3] Zaïane O R, Anthoni M L. Classifying text documents by associating terms with text categories [C] // Zhou X F. Proc of the 13th Australasian Database Conf. Melbourne: Australian Computer Society, 2002: 215-222.
- [4] Agrawal R, Srikant R. Fast algorithms for mining association rules [C] // Bocca J B, Jarke M, Zaniolo C. Proc of the 20th Very Large Data Bases Conference. Santiago, 1994: 487-499.
- [5] Han J, Pei J, Yin Y W. Mining frequent patterns without candidate generation [J]. Data Mining and Knowledge Discovery, 2004, 8(1): 53-87.
- [6] 陈晓云, 陈伟, 王雷, 等. 基于分类规则树的频繁模式文本分类 [J]. 软件学报, 2006, 17(5): 1 017-1 025.
Chen Xiaoyun, Chen Hui, Wang Lei, et al. Frequent pattern text classification based on rules tree [J]. Software, 2006, 17(5): 1 017-1 025. (in Chinese)
- [7] <http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeICTCLAS/> [OL]. 中文自然语言处理开放平台网站, 2006.
<http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeICTCLAS/> [OL]. The Site of Chinese Natural Language Processing Platform, 2006. (in Chinese)

[责任编辑: 顾晓天]

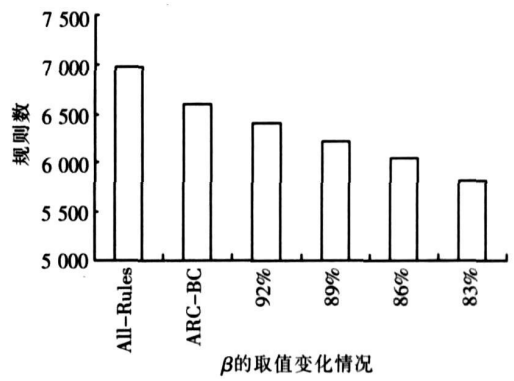


图 3 3 种分类器的规则数比较

Fig.3 Comparison on the number of rules of the three classification algorithms