基于排序的关联分类算法

朱晓燕 宋擒豹

(西安交通大学计算机科学与技术系 西安 710049)

摘 要 提出了一种基于排序的关联分类算法。利用基于规则的分类方法中择优方法偏爱高精度规则的思想和考虑尽可能多的规则,改进了CBA(Classification Based on Associations)只根据少数几条覆盖训练集的规则构造分类器的片面性。首先采用关联规则挖掘算法产生后件为类标号的关联规则,然后根据长度、置信度、支持度和提升度等对规则进行排序,并在排序时删除对分类结果没有影响的规则。排序后的规则加上一个默认分类便构成最终的分类器。选用20个UCI公共数据集的实验结果表明,提出的算法比CBA具有更高的平均分类精度。

关键词 分类,关联规则,排序

中图法分类号 TP31 **文献标识码** A

Classification Mining Using Association Rules Based on Rule Ranking

ZHU Xiao-yan SONG Qin-bao

(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract A new associative classification algorithm based on rule ranking was proposed. The proposed method takes advantage of the optimal rule method preferring high quality rules. At the same time, it takes into consideration as many rules as possible, which can improve the bias of CBA that builds a classifier according to only several rules covering the training dataset. In the proposed algorithm, after the generation of association rules whose consequences are class labels, rules are ranked according to their length, confidence, support, lift and so on. Rules having no influence on the classification result are deleted during ranking. The set of the ranked rules with a default class constructs the final classifier. Finally, 20 datasets selected from UCI ML Repository was used to evaluate the performance of the method. The experimental results show that our method has higher average classification accuracy in comparison with CBA.

Keywords Classification, Association rules, Ranking

1 引言

作为机器学习领域的一个研究热点,分类已引起了研究人员的普遍关注。所谓分类,就是根据给定的训练数据建立分类模型,然后利用这个模型预测新数据对象的类别。

研究人员提出了很多种建立分类器的技术,包括朴素贝叶斯[1]、决策树[2]等。朴素贝叶斯方法是一种基于贝叶斯理论的统计学方法,它可以预测一个新实例属于一个分类的概率。朴素贝叶斯提供了一种简单有效的分类器构造技术,但是它的条件独立假设在现实中往往是不成立的。决策树每次测试一个属性进行分枝,从而构造决策树。分类时,根据决策树决定新实例的分类。但是,每次分枝只测试一个属性,影响了它的分类精度。

最近几年,关联规则的挖掘得到了广泛的研究,主要集中于从数据集中找出有意义的、满足用户需求的关联规则。已经提出了很多关联规则挖掘的有效算法,如经典算法Apriori算法[3]和目前广泛应用的FP-growth算法[4]。近年将关联规则应用于分类的研究取得了较好的效果。这类方法主要对训练集进行挖掘,从而得到一些高质量的规则,然后根据这些规则构建分类器,预测新实例的类标号。CBA[5],MCAR[6]和CMAR[7]都是基于关联规则进行分类的方法。这些方法具有挖掘属性之间高质量的规则,因此可以克服朴素贝叶斯条件独立假设的缺陷,也可以克服决策树每次只测试一个属性进行分枝的不足。

CBA[5]是Liu等人1998年提出的,是最早将关联规则用于分类的算法之一。CBA首先用著名的Apriori算法产生所有后件为类标号的关联规则,然后将这些规则按照优先级进行排序,并选择具有最高优先级的规则集合覆盖训练集,从而构造分类器。这样选出来的规则集合加上一个默认分类构成了最终的分类器。分类时,第一条覆盖新实例的规则决定该实例的分类。

Thabtah等[6]提出的基于关联规则的分类算法MCAR使用最终的规则集中覆盖新实例的一条最优规则预测该实例的分类。MCAR通过存储频繁项集的行号并通过结合这些

到稿日期:2008-09-24 返修日期:2008-12-16 本文受国家自然科学基金项目(编号:60673124),国家"863"计划项目(编号:2006AA01Z183)和教育部"新世纪优秀人才支持计划"项目(编号:NCET-07-0674)资助。

朱晓燕 博士研究生,主要研究方向为可信软件、数据挖掘,E-mail:xyxyzh@gmail.com;宋擒豹 博士,教授,博士生导师,主要研究方向为数据挖掘与软件工程、可信软件。

行号对应的频繁项集产生新的频繁项集，以生成关联规则，从而减少数据库的扫描次数。然后，MCAR 对这些规则进行排序并选择最优的规则构造分类器。与 CBA 不同的是，如果多条规则具有相同的置信度和支持度，则 CBA 随机选择其中的一条，而 MCAR 选择具有高置信度和高代表性的规则。此外，MCAR 选择具有更少条件的规则。

Li 等[7]提出一种基于多关联规则的分类算法 CMAR。CMAR 通过分析规则之间的相关性进行预测。该方法通过建立类关联分布树并扩展 FP-growth 算法来提高关联规则的挖掘效率。分类时，根据加权 χ^2 的值预测新实例的类别。此外，CMAR 采用 CR-树结构来提高关联规则的存储和查找效率。

针对朴素贝叶斯分类器的属性相互独立假设不总是成立的情况，LB[8]利用频繁长项集表示属性间可能存在的依赖关系，扩展了朴素贝叶斯分类器的适用范围。给定一个实例 I，LB 计算每一个分类的条件概率 $P(c_i \mid I)$。获得最大概率的类 c_i 即为分类结果。在极端情况下，该方法找到的频繁项集都是 1-项集，从而退化为朴素贝叶斯分类器。

大量的研究证明，基于关联规则的分类方法总体上可以获得更高的分类精度[5-7,9]。但是这些方法也存在一些不足。CBA 只选用少数几条覆盖训练集的高优先级的规则构造分类器，由此会对新实例的分类结果产生片面的影响。本文旨在克服这种不足，提出了一种新的基于关联规则的分类方法，使用尽可能多的规则构造分类器，从而能更好地预测将来可能出现的新实例。我们设计了相关的实验验证新算法的性能。

本文第 2 节简单介绍 CBA 算法，并指出其存在的不足；第 3 节给出基于排序的关联分类算法，包括算法的简介和各个步骤的详细介绍，以及几个与算法相关的概念；第 4 节进行相关的实验，展示并分析实验结果；最后总结全文。

2 CBA

CBA 是最早将关联规则用于分类的算法之一。CBA 首先产生所有类关联规则 CARs(Class Association Rules)，然后对规则按照优先级进行排序并选择最好的规则覆盖训练集。该算法分两个步骤构造分类器：第一步，发现所有形如 $x_1 \wedge x_2 \wedge \cdots \wedge x_n \rightarrow C_i$ 的关联规则，即后件为类属性值的关联规则。第二步，从已发现的 CARs 中选择高优先度的规则来覆盖训练集，根据这些规则构造具有最小差错率的分类器，从而用于预测新实例的分类。分类时，已构造的分类器中第一条覆盖新实例的规则决定分类结果。

CBA 保证训练集中的每一条实例都被分类器 C 中的一条高优先级的规则覆盖，并且分类器 C 中的每一条规则都至少覆盖训练集中的一条实例。

CBA 的不足之处在于，选用少数几条覆盖训练集的高优先级的规则构造分类器只考虑了训练集中实例的情况。而训练集中的实例只代表将来可能遇见的新实例中的极少数，忽略了剩余规则可能对这些非训练集中的实例产生的影响。ACRR 旨在克服这种不足，保留规则集中可能对分类结果产生影响的所有规则，使得最终构造的分类器在对新实例进行分类时能够得到更高的分类精度。接下来介绍基于排序的关联分类算法，并给出相应的实验，验证该算法的性能。

3 基于排序的关联分类算法

本节首先简单介绍基于排序的关联分类算法的基本思想。在此基础上，依次给出该算法涉及的相关知识和算法各个部分的详细介绍。

3.1 算法简介

本文提出的基于排序的关联分类算法包括如下 3 个部分。

1) 产生规则。使用 Li Wenmin，Han Jiawei 等人[7]提出的改进的 FP-growth 算法产生带类标号的关联规则，并根据设定的最小支持度和最小提升度对产生的规则进行裁剪。

2) 规则排序。根据规则的长度、置信度、支持度、提升度和规则后件出现次数的优先级对规则进行排序，并删除对分类结果没有影响的规则。排序后，优先级高的规则排在前面。选择训练集中出现次数最多的类作为默认分类，加在最终产生的排序规则集合的后面，便构成最终的分类器。

3) 基于规则进行分类。当对新实例 X 进行分类时，从头开始扫描已构造的分类器中的关联规则，找到第一条覆盖 X 的规则，将 X 预测为该规则的后件对应的类标号。如果找不到这样的规则，则将 X 预测为默认分类对应的类标号。

从给定数据集中发现关联规则时，我们采用了带约束的关联规则挖掘算法，目的是为了使得挖掘出来的每一条关联规则的后件都是一个类属性，从而可以直接用于分类。按照规则的长度、置信度、支持度、提升度和规则后件出现的次数确定规则的优先级，将规则按照优先级从高到低进行排序，并删除对分类结果没有影响的规则。排序后的规则加上一个默认分类，便构成最终的分类器。分类时，根据分类器中第一条覆盖新实例的规则或默认分类决定新实例的类标号。具体算法流程如图 1 所示。

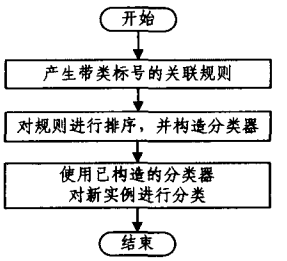

图 1 基于排序的关联分类算法流程图

3.2 相关概念

关联规则的概念由 Agrawal 等人[3]于 1993 年提出，关联规则的挖掘是数据挖掘领域中一个非常重要的研究课题，用来发现数据库中项集之间的关联关系。简单地说，关联规则是用来描述属性之间相互影响程度的。随着大量数据不停地收集和存储，许多业界人士对于从他们的数据库中挖掘关联规则越来越感兴趣。从大量商务事务记录中发现有趣的关联关系，可以帮助许多商务决策的制定，如分类设计、交叉购物和贱卖分析。

设 $I=\{i_1, i_2, \cdots, i_m\}$ 是项的集合。设任务相关的数据 D 是数据库事务的集合，其中每个事务 T 是项的集合，使得 $T \subseteq I$。每一个事务有一个标识符，称作 TID。设 A 是一个项集，事务 T 包含 A 当且仅当 $A \subseteq T$。关联规则是形如 $A \Rightarrow B$ 的蕴涵式，其中 $A \subset I$，$B \subset I$，并且 $A \cap B = \Phi$。对于关联规则

$A \Rightarrow B$

支持度 $Support(A \Rightarrow B) = P(A \cup B)$ (1)

置信度 $Confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)}$ (2)

提升度 $Lift(A \Rightarrow B) = \frac{support(A \cup B)}{support(A) \times support(B)}$ (3)

Support 是 D 中事务同时包含 A 和 B 二者的百分比，是对关联规则重要性的衡量，说明该规则在所有事物中有多大的代表性。支持度越大，关联规则越重要。如果项集满足最小支持度 *min_sup*，则称它为频繁项集。

Confidence 是指 D 中包含 A 的事务同时也包含 B 的百分比，是对关联规则准确度的衡量。进行预测时，*Confidence* 是一个很自然的选择，它反映了在给定 A 的前提下 B 发生的后验概率。

Lift 有时也称为 *Interest*，它是 A 和 B 同时发生的概率与在假定 A 和 B 独立的前提下 A 和 B 同时发生的概率之间的比值。*Lift* 用来衡量 A 和 B 之间的关联与 A 和 B 相互独立偏离的程度。如果 *Lift* 接近于 1，A 和 B 就是独立的；如果 *Lift* 小于 1，这条规则就是没有很大意义的。*Lift* 越大，规则的实际意义就越好。

3.3 规则排序及分类器构造

我们使用改进的 FP-growth 算法产生带类标号的关联规则，并根据设定的最小支持度和最小提升度对产生的规则进行裁剪。在此不进行详细介绍。下面主要介绍规则排序及分类器构造，以及如何使用构造的分类器进行分类。本节介绍规则排序及分类器构造。下一节介绍如何使用构造的分类器进行分类。

在介绍具体的算法之前，我们先给出“≥(优先级高于)”的定义。

我们称规则 r_i 的优先级高于规则 r_j 的优先级，记作 $r_i \geqslant r_j$，如果满足：

r_i 的长度大于 r_j；或者 r_i 和 r_j 长度相等，但是 r_i 的置信度高于 r_j 的置信度；或者 r_i 和 r_j 的长度和置信度相等，但是 r_i 的支持度高于 r_j 的支持度；或者 r_i 和 r_j 的长度、置信度和支持度相等，但是 r_i 的提升度高于 r_j 的提升度；或者 r_i 和 r_j 的长度、置信度、支持度和提升度相等，但是 r_i 的后件出现的次数大于 r_j 的后件出现的次数。如果 r_i 和 r_j 的以上各个参数都相等，则产生较早的规则具有较高的优先级。

下面我们给出 ACRR 对规则进行排序的算法，如算法 1 所示。

算法 1 ACRR 的排序算法

```
Input: rule set R
Output: ranked rule set R
Step 1: for i = 0 to R. size do
Step 2:   temp = i
Step 3:   for j = i+1 to R. size do
Step 4:     if r_temp ≥ r_j then
Step 5:       if r_temp cover r_j then
Step 6:         delete r_j from R
Step 7:         j--
Step 8:       end if
Step 9:     else if r_j cover r_temp then
Step 10:          delete r_temp
Step 11:          j--
Step 12:        end if
Step 13:        temp = j
Step 14:    end if
Step 15: end for
Step 16: if temp ! = i then
Step 17:   r_i ↔ r_temp
Step 18: end for
```

排序算法采用选择排序的思想，每一次循环找出优先级最高的规则，并删除对分类结果没有影响的规则。

经过该排序算法之后，规则集中的所有规则都按照长度、置信度、支持度、提升度和后件出现次数进行排序，优先级越高的规则排在越前面，分类时选择该规则的概率就越大。

选用训练集中出现次数最多的分类 default_class 作为默认分类，在找不到覆盖新实例的规则时，便使用默认分类作为新实例的类标号。排序后的规则加上默认分类便构成了最终的分类器，可以用于对新实例的预测。

3.4 使用构造的分类器进行分类

当对新实例 X 进行分类时，从头开始扫描排序后的关联规则集合，找到第一条覆盖 X 的规则，将 X 预测为该规则的后件对应的分类。如果找不到这样的规则，则将 X 预测为默认分类对应的类标号。

下面我们给出如何使用已构造的分类器对新实例进行分类的算法，如算法 2 所示。

算法 2 分类算法

```
Input: ranked rule set R
       default class label default_class
       X-the new instance to be classified
Output: the class label C of X
Step 1: for each rule r_i ∈ R
Step 2:   if r_i cover X then
Step 3:     C←consequence(r_i)
Step 4:     return
Step 5:   end if
Step 6: end for
Step 7: C←default_class
```

从上面的讨论可以看出，ACRR 保留了所有可能对新实例的分类产生影响的规则。当 CBA 选用一条规则 r 的时候，要求 r 必须至少覆盖当前训练集中剩余实例中的一条，这条实例没有被比 r 优先级高的规则 r' 覆盖(被 r' 覆盖的规则在选用 r' 时就已经从训练集中删除了)。训练集相对于以后可能遇到的新实例来说，只代表了极少数的情况。而没有满足该要求的规则，在对新实例分类的时候可能会比分类器中的规则得到更精确的分类结果。

CBA 只根据训练集中数据的情况就把这些规则删除了。ACRR 对此进行了改进，保留了所有可能对新实例的分类产生影响的规则，从而能够得到精度更高的分类器。下一节我们通过实验对 ACRR 的性能进行验证。

4 实验及结果分析

为了测试 ACRR 的分类精度，我们进行了实验，并给出了实验结果。结果表明，相对于 CBA，ACRR 具有更高的平均分类精度。

4.1 实验设置

1) 设置

ACRR 和 CBA 的所有实验都是在 1G 内存，运行 Windows XP 的 Pentium PC 上进行的。ACRR 是基于 Weka[10] 平台开发的，CBA 使用文献[5]的作者给定的程序版本。

2) 测试数据

为了测试本文提出算法的性能，我们选用数据挖掘领域用来比较不同算法性能的标准数据 UCI ML Repository[11] 中的 20 个数据集进行实验。

3) 验证方法

为了充分利用实验数据，对给定的数据集采用十折交叉验证。即将数据集分成 10 份 $S_1, S_2, \cdots, S_{10}$，训练和测试进行 10 次。在第 i 次迭代，S_i 用作测试集，其余的子集都用于训练分类算法。

4)) 度量标准

本文采用分类精度衡量不同算法的性能，分类精度定义如下：

$$\text{分类精度}=\frac{\text{正确预测的实例数}}{\text{预测的总实例数}}\times 100\%$$

5)基准方法

为了更好地分析本文提出算法的性能，我们将其与经典的基于关联规则的分类算法 CBA 进行对比，看分类精度是不是得到了提高。

4.2 实验及结果分析

对于 CBA，我们将最小支持度设为 1%，最小置信度设为 50%，其他的参数使用文献[5]中的默认值。

对于 ACRR，我们将最小支持度设为 1%，最小提升度设为 1.2，并使用 weka 给定的离散化方法对连续性属性进行离散化。

表 1 给出了试验中使用的 20 个数据集的相关属性，以及 ACRR 和 CBA 在每个数据集上的分类精度和对 20 个数据集的平均分类精度。黑体数值表示两个算法对一个数据集分类的最高精度。

从表 1 可知：1) ACRR 比 CBA 平均分类精度提高了 16.96%；2)在 20 个数据集中，本文提出的算法在 14 个数据集上比 CBA 具有更高的分类精度，CBA 仅在 6 个数据集上比 ACRR 的分类精度高。3)在 ACRR 分类精度低于 CBA 分类精度的 6 个数据集中，平均只降低了 3.14%。

表 1 ACRR 和 CBA 分类精度对比

数据集	属性个数	实例个数	类属性取值个数	分类精度	
				CBA	ACRR
1 alance-scale	5	625	3	71.40	**88.16**
2 balloons	5	16	2	15.00	**87.50**
3 car	7	1728	4	**86.91**	85.19
4 lenses	5	24	3	**85.00**	83.33
5 tic-tac-toe	10	958	2	77.88	**95.09**
6 nursery	9	556	2	**95.40**	88.48
7 pima	9	768	2	73.93	75.26
8 tae	6	151	3	15.32	**47.02**
9 haberman	4	306	2	71.55	**74.18**
10 glass	10	214	7	72.46	**79.91**
11 breast	10	699	2	**95.99**	92.70
12 iris	5	150	3	**94.67**	93.33
13 led7	8	3200	10	69.50	**73.91**
14 diabetes	9	768	2	74.84	**76.30**
15 cmc	10	1473	3	35.02	**48.74**
16 ecoli	8	336	8	72.26	**83.63**
17 liver-disorder	7	345	2	56.98	**63.19**
18 post	9	90	3	**54.46**	53.33
19 yeast	9	1484	10	38.60	**55.92**
20 king	7	28056	18	0.89	**26.19**
平均分类精度				62.90	**73.57**

综上所述，本文提出的分类算法 ACRR 比 CBA 具有更高的平均分类精度，并且在 70%的数据集上比 CBA 的分类精度更高。

结束语 本文提出了一种新的关联分类算法，旨在改进 CBA 只根据几条覆盖训练集的规则构造分类器的不足，从而得到分类精度更高的分类器。本算法首先采用关联规则挖掘算法产生后件为类标号的关联规则；然后根据规则的长度、置信度、支持度和提升度等对规则进行排序，使得高优先级的规则排在前面。排序后的规则加上一个默认分类便构成了最终的分类器；对新实例进行分类时，选用第一条覆盖新实例的规则预测该实例的类标号。如果找不到这样的规则，则将默认分类作为新实例的类标号。

为了验证算法的有效性，我们选取 20 个 UCI 数据集进行了实验。将本文提出的算法同 CBA 进行对比。结果表明，相对选定的数据集而言，提出的方法比 CBA 具有更高的平均分类精度。

参 考 文 献

[1] Duda R O, Hart P E. Pattern Classification and Scene Analysis [M]. New York: Wiley-Interscience, 1973

[2] Quinlan J R. C4. 5: Programs for machine learning[M]. San Mateo, CA: Morgan Kaufmann, 1993

[3] Agrawal R, Imilinski T, Swami A. Mining Association Rules Between Sets of Items in Large Database[C] //Proceedings of the ACM SIGMOD Conference on Management of Data. Washington DC, 1993: 207-216

[4] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C] //Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, TX, 2000: 1-12

[5] Liu B, Hsu W, Ma Y. Integrating Classification and Association Rule Mining[C] //Proceedings of the Fourth ACM SIGKDD Ubtebatuibak Conference on Knowledge Discovery and Data Mining. New York, 1998: 80-86

[6] Thabtah F, Cowling P, Peng Y. MCAR: multi-class classification based on association rule[C] //Proceedings of the Third ACS/IEEE International Conference on Computer Systems and Applications. 2005: 33-39

[7] Li W, Han J, Pei J. CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules[C] //Proceedings of the 2001 IEEE Int. Conf. on Data Mining (ICDN 2001). San Jose, California, 2001: 369-376

[8] Meretakis D, Wuthrich B. Extending Naive Bayes Classifiers Using Long Itemsets[C] //Proceedings of KDD-99. San Diego, USA, 1999

[9] Yin X, Han J. CPAR: classification based on predictive association rules[C] //Proceedings of the Third SIAM International Conference on Data Mining (SDM'03). May 2003

[10] WEKA: Data Mining Software in Java[EB/OL]. http://www.cs.waikato.ac.nz/ml/weka

[11] Merz J C, Murphy, et al. UCI repository of machine learning databases [EB/OL]. http://www.ics.uci.edu/mlearn/-MLRepository.html, 1997