基于遗传算法的分类规则序列生成

刘海卫[1],倪恩志[2],周昌乐[2]*
(1. 厦门大学软件学院,2. 厦门大学智能科学与技术系,福建 厦门 361005)

摘要: 与当前常用的分类方法相比,遗传算法具有较强的伸缩性和全局搜索能力,易于并行计算等优点. 但通过遗传算法得到的一组分类规则之间常常存在冲突. 本文先将分类规则表示成二进制编码,采用 F-measure 作为适应度评估函数,并设计了有效的杂交,变异等遗传算子,使遗传算法适合用在分类规则挖掘中. 在遗传算法中增加了冲突解决机制,并结合顺序覆盖算法,使之可以得到分类规则的序列,解决了规则间的冲突,形成了一个完整的分类方法. 最后针对具体实例作了测试,并将实验结果与分类算法 J4.8 得到的结果进行了对比,表明该方法略优于 J4.8.

关键词: 分类;数据挖掘;遗传算法
中图分类号: TP 311 **文献标识码**: A **文章编号**: 0438-0479(2008)02-0202-05

分类方法[1]是数据挖掘中的一个重要研究方向. 分类就是根据数据集的特点找出类别的概念描述,这个概念描述代表了这类数据的整体信息,也就是该类的内涵描述,并使用这种类的描述对未来的测试数据进行分类. 目前分类的主要方法有:决策树方法、神经网络方法、贝叶斯方法、粗糙集方法和遗传算法等.

其中遗传算法(Genetic algorithm, GA)[2]是一种全局优化算法,具有较强的伸缩性. 在数据挖掘任务中,我们所要处理的数据常常非常庞大,所以遗传算法很适合在这种场合使用[3]. 并且与其他算法相比,遗传算法能更好地处理属性间的交互效应[4].

到目前为止,对遗传算法在分类规则挖掘中的研究主要集中在规则的个体编码表示、适应度函数的选择、规则兴趣度的评估[5]和规则约束[6]等方面. 通过对这些方面的研究使得遗传算法能够挖掘出用户感兴趣且易于理解的分类规则. 如果得到的是多个规则,这些规则之间常常是有冲突的,它们的使用顺序对分类结果有很大的影响. 然而在很多文献中只是关注如何得到这些规则,而对如何将得到的规则形成一个完整的分类方法还有欠研究. 如在文献[7]中只是提到如何生成分类规则,在文献[8]中为每个目标属性生成一条规则,事实上这些规则之间会存在着很多冲突,在这些文章中并未提及如何解决这些冲突,以及在具体分类中应用这些规则. 本文提出的算法能够得到一个分类规则的序列,这个序列是有序的,这样就消除了规则间的冲突,得到了一个完整的分类方法.

收稿日期:2007-07-11
*通讯作者:dozero @xmu.edu.cn

1 分类规则

1.1 分类规则的表示

在一些经典的分类算法(如 ID3, C4.5, J4.8)中,采用的是基于决策树的方法,即使用树形结构来表示分类知识. 决策树的每个内部节点代表一个属性,每个分支是一个对于节点属性的判定表达式,而叶子节点表示类别. 使用这种方法来进行分类比较直观,然而在遗传算法中很难使用这个结构来进行操作(如杂交、变异等). 因此需要有更简单的分类知识表示方式,其中应用最多的就是分类规则.

分类规则使用 IF-THEN[9]的形式描述,如下所示:

IF〈选定属性满足条件〉THEN〈属于某个类别〉.

其中,IF 部分称为前件,THEN 部分称为后件. 我们先来讨论前件的表示. 假设一个属性具有 k 个离散的值,我们可以用 k 个二进制位来表示它. 每个值对应一个二进制位,该位为 1,表示属性为这个值,如果有两个以上的位为 1,表示这个属性可以为这些值中的任意一个. 举个例子,有一个属性为"结婚状况",它有 4 个值:"未婚"、"已婚"、"离婚"和"丧偶",用 IF-THEN 形式描述"已婚或离婚"如下:

IF (结婚状况 = "已婚" OR "离婚").

若用二进位,则表示为"0110". 具有连续值的属性也可以用二进制表示为适合遗传算法处理的形式,具体可参考文献[8].

后件的表示可有 3 种形式[10]. 第一,后件采用与前件同样的编码;第二,每个前件和某一类别相联系,

在运行过程中始终不变;第三,动态选择最适合前件的类别作为后件.其中第 3 种能够使规则个体的准确率最大化,而且能提高算法的效率[7],所以本文也采用这种方法表示.

1.2　分类规则间的冲突

通过遗传算法得到的分类规则之间常常存在着冲突.例如可能同时得到以下两条规则:

●IF 天气 = 晴朗 Then 适合户外运动;

●IF 天气 = 晴朗 and 气温 = 很高 Then 不宜户外运动.

很显然这两条规则是有冲突的,但是只要使用次序得当就能解决这个问题.比如我们先使用第 2 条规则,如果符合条件则输出"不宜户外运动",否则,再接着使用第 1 条规则.这样就避免了它们之间的冲突,在现实中也合情合理.所以关键是规则的次序问题.

在文献[1]中提到两种冲突解决策略.第 1 种是根据规则前件的属性数目来排序,越是严苛的前件排得越靠前.第 2 种是根据规则来排序,这又可以分为:基于类别的排序和基于规则的排序.前者是根据类别出现的数目来排序的,在测试集中数目越多的类别,排得越靠前.后者是根据符合规则的数据量来排序的,在测试集中越多的数据与规则匹配,它排得越靠前.

本文使用的算法得到的分类规则序列与最后一种策略有点类似.它们之间的不同是本文不是单单考虑了规则匹配的数据多少,而是综合了准确率、覆盖率等指标,使得具有最大适应度的规则最先输出,排在规则序列的最前面.因此这样得到的序列更为合理和有效.

2　算法描述

2.1　算法的总体描述

IF-THEN 形式的规则可以使用顺序覆盖算法直接从训练集中得到.这个算法已经被广泛的应用于分类规则的提取.在文献[11]中它和遗传算法结合,用在分类规则的提取,并在一些具体的任务中(monk-1, car-evaluation, breast-cancer)取得了较好的结果.在这里我们通过这种方法解决分类规则间的冲突问题.该算法有多种不同的改进版本,如 AQ[12],CN2[13]和 RIPPER[14]等.算法的基本描述如下所示.

```
Sequential_Covering(Training_Dataset){
LearnedRules = {};
Rule = Learn_One_Rule(Training_Dataset);
While (not Terminating condition) {
  LearnedRules = LearnedRules + Rule;
  Training_Dataset = Training_Dataset-data covered by Rule
  Rule = Learn_One_Rule(Training_Dataset);
}
}
```

算法每次学习到一条规则,然后删除训练集中被该规则覆盖的数据,继续循环,直到满足终止条件为止.我们把终止条件设为它已经无法得到新的规则.这包括 2 种情况:第一,训练集中剩下的数据都属于同一类别;第二,最后得到的规则已经不满足我们设定的最低适应度(适应度的设计在下文描述)要求.算法最后输出的是一个分类规则的序列,在这个序列中不存在冲突,所以可以直接用于分类.关于该算法解决冲突的机制,我们在 2.3 中进行介绍.

2.2　Learn_One_Rule 的遗传算法实现

2.2.1　遗传算子的设计

在这一部分中,我们详细描述遗传算法各种算子的实现.

算法采用的选择策略是从当前的个体空间(假定空间大小为 p)内选择一定数量(假定为 $k, k < p$)的个体放到下一代个体空间.在选择的时候,适应度越大的个体,被选到的机会也会更大.

杂交算子采用的是两点杂交,即随机生成两个交换点,将父体 A 在这两个交换点中间的基因片段和父体 B 在这之外的片段组合生成一个新个体放到子代中,同样将父体 B 在这两个交换点中间的基因片段和父体 A 在这之外的片段组合生成另一个新个体.除此之外,算法还采用了换位杂交的杂交算子.换位操作就是将两个父体在某个位置上的基因相互交换生成两个新个体.通过杂交应获得 $p - k$ 个个体放到下一代空间中.

变异策略是随机地选择 m 个个体,然后翻转基因中随机选择的某个比特位.另外,根据分类规则的特点,在算法中加入插入/删除属性的操作,即随机选取一个属性,将这个属性的比特位设为全 1 或是只有一个位为 1.加入这个操作后,使算法可以在更大范围内搜索分类规则,从 IF 部分包含一个属性到多个属性的规则中选择最有效的分类规则.

适应度规则对算法的性能和结果有很重要的影响.算法的目标是得到适应度最大的个体,同时这个个体所表示的分类规则也应该是最有效的.所以必须根据分类规则的评估标准来确定适应度规则.

分类规则最简单的评估标准是准确率,但准确率在许多情况下并不能符合人们实际的需要.假定有一个分类法将医疗数据分类为 cancer 或 non-cancer.90 %的准确率使得该分类法看上去相当准确,但是如果实际只有 3 %～4 %的训练样本是 cancer,则 cancer 样本被正确分类的可能性就很小了.因此准确率并不

能很好地评估分类方法.以下给出几个更有效的评估标准.

假定样本具有两个类别[15],分别为 yes 和 no.将属于 yes 的样本称为正样本,属于 no 的样本称为负样本.分类后的结果如表 1 所示.

表 1 两个类别样本的分类

Tab.1 Different outcomes of a two-class prediction

实际的分类	预测的分类	
	yes	no
yes	True Positive	False Negative
no	False Positive	True Negative

其中,True positive(TP)表示被正确分类为 yes 的样本数;False positive(FP)表示被分类为 yes 而实际上是 no 的样本数;False negative(FN)表示被分类为 no 而实际上是 yes 的样本数;True negative(TN)表示被正确分类为 no 的样本数.基于以上提到的 4 个值,可以有以下几种度量:TP rate,FP rate,Precision 和 F-Measure.

$$\text{TP rate} = \frac{TP}{TP + FN},\ \text{FP rate} = \frac{FP}{FP + TN},$$

$$\text{Precision} = \frac{TP}{TP + FP},$$

$$\text{F-measure} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}.$$

其中除 FP rate 外,其他 3 个都是值越大越好,并且每种度量有各自的侧重,因此在评估时可同时使用.而 F-measure 度量是较为平衡的一种度量,因此将它作为适应度评估函数是合适的.

2.2.2 遗传算法描述

以下对算法作一个简单的描述,如下所示:

```
Learn_One_Rule(Training_Dataset){
  Randomly Generate a new generation;
  While(current generation < max generation){
    Select k individuals out of population to add to new population;
    Crossover (p - k)/2 pairs of individuals to produce p - k individuals to add to new population;
    Mutate m individuals in new population;
    Save the fittest individual;
    population = new population;
    Solve_Conflict(population);
  }
  Output the fittest individual;
}
```

算法首先随机生成一代新的个体,然后执行这样的循环:根据我们之前确定的选择策略,即个体适应度越大选择的概率也越大,从当前群体中选择 k 个个体放到下一代群体中;选择 $(p - k)/2$ 对的个体,对每对进行杂交,产生 $p - k$ 个个体,并放到下一代群体中;在下一代群体中,随机选择 m 个个体进行变异;保存具有最优适应度的个体;将下一代群体替换为当代群体,并执行冲突解决函数(在 2.3 中介绍).循环直到我们事先规定的最大代数,然后输出具有最优适应度的个体.

2.3 冲突解决机制

在 1.2 中我们给出了一个规则冲突的例子.对于这种冲突的解决描述如下:

假设在某代群体中存在这样两条规则,分别记作 $R1$ 和 $R2$,它们满足:在数据集 D 中满足 $R1$ 前件的数据集合为 $D1$,满足 $R2$ 前件的数据集合为 $D2$,$D1 \supset D2$;$R1$ 和 $R2$ 的后件分别为类别 $T1$ 和类别 $T2$,$T1 \neq T2$.不失一般性的假设数据集中的类别分布如图 1 所示.

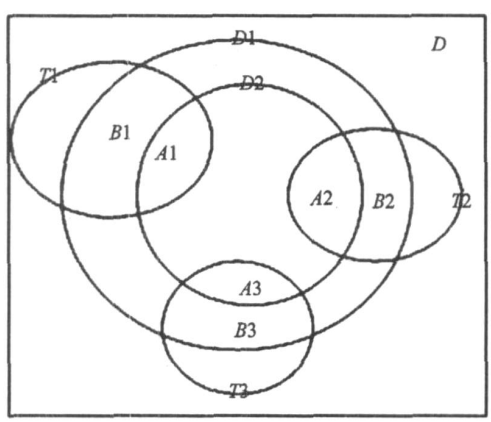

图 1 数据集类别分布

Fig.1 Distribution of categories in dataset

如果使用 $R1$ 分类,则分类错误的数据为 $A2 + B2 + A3 + B3$.因为 $D1 \supset D2$,所以经过 $R1$ 分类后,数据集中就不存在符合 $R2$ 前件的数据.

如果先使用 $R2$,再使用 $R1$ 分类,则分类错误的数据为 $A1 + B2 + A3 + B3$.

比较可知,当 $A1 > A2$ 时,使用第 1 种方法分类,能提高准确率;否则,应使用第 2 种方法分类.因为 $R2$ 的 $\text{Precision} = \frac{TP}{TP + FP} = \frac{A2}{A2 + (A1 + A3)}$,所以只要满足 $\text{Precision} > \frac{1}{2}$,就必定满足 $A1 < A2$,此时要使用第 2 种方法.

表 2　GA 的测试结果

Tab.2　The results of GA

TPr	FPr	Precision	FM	Class
1.00	0	1.00	1.00	diaporthe-stem-canker
1.00	0	1.00	1.00	charcoal-rot
1.00	0	1.00	1.00	rhizoctonia-root-rot
1.00	0	1.00	1.00	phytophthora-rot
1.00	0	1.00	1.00	brown-stem-rot
1.00	0	1.00	1.00	powdery-mildew
1.00	0	1.00	1.00	downy-mildew
0.75	0.05	0.78	0.76	brown-spot
1.00	0.003	0.91	0.95	bacterial-blight
0.80	0	1.00	0.89	bacterial-pustule
1.00	0	1.00	1.00	purple-seed-stain
0.875	0	1.00	0.93	anthracnose
0.60	0	1.00	0.75	phyllosticta-leaf-spot
0.76	0.05	0.76	0.76	alternarialeaf-spot
0.75	0.01	0.95	0.84	frog-eye-leaf-spot

注:TPr 为 TP rate;FPr 为 FP rate;FM 为 F-Measuer.

表 3　J4.8 的测试结果

Tab.3　The results of J4.8

TPr	FPr	Precision	FM	Class
1.00	0.002	0.952	0.976	diaporthe-stem-canker
1.00	0	1.00	1.00	charcoal-rot
0.95	0.004	0.905	0.927	rhizoctonia-root-rot
0.6	0.004	0.6	0.6	phytophthora-rot
1.00	0	1.00	1.00	brown-stem-rot
1.00	0	1.00	1.00	powdery-mildew
1.00	0	1.00	1.00	downy-mildew
0.913	0.013	0.933	0.923	brown-spot
1.00	0.006	0.87	0.93	bacterial-blight
0.85	0	1.00	0.919	bacterial-pustule
1.00	0	1.00	1.00	purple-seed-stain
0.955	0	1.00	0.977	anthracnose
0.65	0.008	0.765	0.703	phyllosticta-leaf-spot
0.934	0.042	0.817	0.872	alternarialeaf-spot
0.769	0.029	0.843	0.805	frog-eye-leaf-spot

注:TPr 为 TP rate;FPr 为 FP rate;FM 为 F-Measuer.

在遗传算法描述中,Solve_Conflict 函数执行的功能就是判断两个个体之间是否满足上述 $R1$, $R2$ 的关系,如果满足,接着判断 $R2$ 的准确度. 如果大于$\frac{1}{2}$,则将 $R1$ 的适应度置 0,使得 $R2$ 可能在本次循环中胜出;如果小于$\frac{1}{2}$,则将 $R2$ 的适应度置 0,使得 $R1$ 可能在本次循环中胜出. 这样就能在保证准确率的同时得到一个没有冲突的序列,达到我们最初的目的.

3　测试与比较

这里使用 soybean 数据集进行测试,并与 J4.8 算法的测试结果进行比较. 在 soybean 中总共有 35 个属性,19 个类别,在测试中只选用了 14 个类别的样本. 由于数据集属性较多,根据经验包含较多属性的规则或者准确率很低,或者虽然准确率高但覆盖样本较少,因此效果不佳. 因此,设定规则中的最大属性个数为 6. 各种参数设定如下:选择概率:15 %,换位概率:40 %,变异概率:50 %,插入/删除属性概率:50 %,最大属性个数:6,个体数目:100,演化代数:500. 最终得到的结果如表 2 所示,J4.8 算法的结果如表 3 所示.

从以上数据来看,GA 的实验结果略优于 J4.8,这说明通过 GA 挖掘出的分类规则具有较好的分类能力.

4　结论与展望

分类系统被人们越来越多地应用在科学、工程和经济领域中,是目前遗传算法研究中一个十分活跃的领域. 把遗传算法应用于分类规则的挖掘,需要解决以下几个问题:(1) 设计合适的编码方式,使分类规则可以表示成个体,并且这个编码必须适合于进行杂交,变异等遗传算子的操作;(2) 设计有效的适应度评估策略,使得到的分类规则在保证覆盖率的同时能够准确地分类;(3) 需要一种冲突解决策略,解决多个规则之间的冲突. 针对第 1 个问题,本文采用 if-then 形式表示分类规则,并用二进制编码. 这样的编码方式很适合遗传算子的操作;对于第 2 个问题,本文分析了 TP rate,FP rate,Precision 和 F-Measure 这 4 种度量方式,认为 F-Measure 是一种较为平衡的度量,以此作为适应度评估的标准是合适的;最后,结合顺序覆盖算法,调用遗传算法进行分类规则序列的生成,解决了规则间的冲突,使得挖掘到的分类规则形成一个完整的分类方法. 实验结果显示,遗传算法挖掘到的分类规则具有较好的分类能力.

本文测试用到的数据集较小. 由于 GA 有更强的伸缩性和全局搜索能力,能够更好地处理属性间的交互作用,所以本文中提到的算法在大型数据库的分类规则挖掘中更能体现它的优势. 下一步的工作是在大型数据库中对本算法进行测试,并做出改进以充分发挥遗传算法的优势.

参考文献:

[1] Han Jiawei ,Micheline Kamber. Data mining :concepts and techniques[M]. 2nd ed. Beijing: China Machine Press , 2006.

[2] 潘正君 ,康立山 ,陈敏屏. 演化计算[M]. 北京 :清华大学出版社 ,1998.

[3] Smith S F. A learning system based on genetic algorithms [D]. Pittsburgh : University of Pittsburgh ,Department of Computer Science ,1980.

[4] Freitas A A. Understanding the crucial role of attribute interation in data mining[J]. Artificial Intelligence Review ,2001 ,16(3) :177 - 199.

[5] Freitas A A. On objective measures of rule surprisingness [C]//Lecture Notes in Artificial Intelligence 1510 : Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery. Nantes , France : Springer-Verlag ,1998 :1 - 9.

[6] Chiu Chaochang , Hsu Peilun. A constraint-based genetic algorithm approach for mining classification rules [J]. IEEE Transactions on Systems , Man , and Cybernetics , Part C: Applications and Reviews , 2005 , 35 (2) : 205 - 220.

[7] Freitas A A. A genetic algorithm for generalized rule induction[C]//Advances in Soft Computing-Engineering Design and Manufacturing. London : Springer-Verlag , 1999 : 340 - 353.

[8] Romao W , Freitas A A , Gimenes Itana M S. Discovering interesting knowledge from a science and technology database with a genetic algorithm[J]. Applied Soft Computing ,2004 ,4 :121 - 137.

[9] De Jong K A ,Spears W M , Gordon D F. Using genetic algorithms for concept learning [J]. Machine Learning , 1993 ,13 :161 - 188.

[10] Freitas A A. A survey of evolutionary algorithms for data mining and knowledge discovery[C]//Advances in Evolutionary Computation. London : Springer-Verlag ,2001.

[11] Weijters Ton , Paredis Jan. Genetic rule induction at an intermediate level [J]. Knowledge-Based Systems ,2002 , 15(1/2) :85 - 94.

[12] Michalski R S. On the quasi-minimal solution of the general covering problem[C]//Proceedings of the V International Symposium on Information Processing. Bled , Yugoslavia : Elsevier ,1969 :125 - 128.

[13] Clark P , Niblett T. The CN2 induction algorithm [J]. Machine Learning ,1989 ,3 :261 - 283.

[14] Cohen W. Fast effective rule induction[C]//Proceedings of the Twelfth International Conference on Machine Learning (ICML '95). Tahoe City , CA : Morgan Kaufmann ,1995 :115 - 123.

[15] Written Ian H , Eibe Frank. Data mining : practical machine learning tools and techniques [M]. Beijing : China Machine Press ,2005.

Generating a Sequence of Classification Rules with a Genetic Algorithm

LIU Hai-wei[1] ,NI En-zhi[2] ,ZHOU Chang-le[2]*

(1. School of Software ,Xiamen University ,2. Department of Cognitive Science ,Xiamen University ,Xiamen 361005 ,China)

Abstract : Compared with common classification algorithms , genetic algorithm (GA) had more powerful flexibility and global searching capability. However ,there were many conflicts among classification rules discovered by GA. In this paper ,classification rules were represented by binary codes. F-measure was used as fitness evaluation. We also designed efficient crossover ,mutation operators. Moreover ,solving conflict function was integrated with GA and sequential covering algorithm was combined with GA to get a sequence of classification rules. This approach turned out to be a solution to conflicts among rules and formed a complete classifying method. As could be seen in our experiment ,the result of the algorithm designed in this paper could be proved to be better than that of J4.8.

Key words : classification ; data mining ; genetic algorithms