基于集成学习的覆盖算法

贾瑞玉,冯伦阔,李永顺,张新建
(安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039;
安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:介绍了传统的领域覆盖算法和交叉算法,并分析它们各自存在的缺点以及造成这些缺点的原因。针对传统的领域覆盖算法存在的泛化能力不足以及交叉覆盖算法存在的正确率不高的问题,提出了一种新的改进算法——基于集成学习的覆盖算法(CABE)。CABE是利用集成学习来整合交叉覆盖算法和领域覆盖算法,是通过对领域覆盖算法中的拒识样本的处理来提升算法的精度。使用UCI数据集进行实验,实验结果表明,改进的算法提高了算法分类的精度。

关键词:覆盖算法;集成学习;交叉覆盖;集成覆盖
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2009)07-0076-04

Cover Algorithm Based on Ensemble Learning

JIA Rui-yu, FENG Lun-kuo, LI Yong-shun, ZHANG Xin-jian
(Ministry of Education Key Laboratory of Intelligent Computing & Signal Processing,
Anhui University, Hefei 230039, China;
School of Computer Sci. & Tech., Anhui Univ., Hefei 230039, China)

Abstract: Introduces the traditional CA(Cover Algorithm) and the CCA (Cross Cover Algorithm), and analyses the shortcomings of them and the reasons. For low generalization ability of CA and low accuracy of CCA, propose an improved algorithm - CABE(Cover Algorithm Based on Ensemble). CABE integrates CA and CCA through ensemble learning to improve performance through dealing with those samples that can't be classified. Finally the experimental results show that the CABE can improve the accuracy of classification.

Key words: cover algorithm; ensemble learning; cross cover algorithm; ensemble cover algorithm

0 引 言

张铃等人在文献[1~3]中提出了领域覆盖算法(Cover Algorithm, CA)和交叉覆盖算法(Cross Cover Algorithm, CCA),它们都是根据神经网络的几何意义提出的,前者利用M-P神经元模型的几何意义,得出一个领域覆盖的设计原则,后者是针对前者构造的超球分类器只具有局部性质的不足而提出的。近些年来,在领域覆盖算法和交叉覆盖算法基础上,又做了大量的改进工作。文献[4]是将SVM中的核函数法与构造性学习的覆盖算法相融合,给出一种新的核覆盖算法。文献[5]是将样本集分成几个部分,对各部分分别选择一组适应它们的特征子集。文献[6]将覆盖算法扩展成核覆盖算法(以高斯函数为核函数),再利用高斯函数的概率意义(高斯分布),为核覆盖算法建立一个有限混合概率模型并利用"最大似然原理"引入全局优化计算,最终将覆盖从确定的模型扩展成概率模型。同时,领域覆盖算法及交叉覆盖算法在实际应用中也取得了许多进展,文献[7~9]分别是领域覆盖算法及其改进算法在文本分类、煤价预测、图像识别领域的具体应用,并且都有较好的结果。

虽然领域覆盖算法和交叉覆盖算法在实验和应用中表现出很强性能,但两者都有自己的缺点。文中利用集成学习将交叉覆盖和领域覆盖结合起来,提出基于集成学习的覆盖算法,它是通过对拒识样本的处理,从而达到性能的提升,并通过实验对算法的有效性进行了验证。

1 领域覆盖算法和交叉覆盖算法

1.1 领域覆盖算法

在文献[1]中张铃等人根据神经网络的几何意义提出了领域覆盖算法。该算法的主要思想为:设一给定的输入集K,设K分为k子集,即这些样本的输出有k

收稿日期:2008-11-10;**修回日期**:2009-01-04
基金项目:安徽省自然科学研究项目(kj2008B092)
作者简介:贾瑞玉(1965-),女,副教授,研究方向为计算机图形学、数据挖掘、人工智能。

种,另设一个集合 $I(t)$ 来记录对应 k 子集里样本的标号。在这种情况下,若能够取一批"球形领域"C_j(t 表示子集号,j 表示每个子集被分为若干个领域),对于 $C = \bigcup C_j$ 使得它只覆盖 j 属于 $I(t)$ 的样本,而不覆盖 j 不属于 $I(t)$ 的样本,且 C 互不相交,那么这样就完成了分类要求。学习中,构造 C_j 的方法是[1]:

第一步:任取一个第 t 类尚未被覆盖的点 a;

第二步:找出离 a 最近的异类点 b,两者距离为 d_1;

第三步:找出离 a 最远且距离小于 d_1 的同类点 c,两者距离为 d_2;

第四步:覆盖领域的半径为 $r = (d_1 + d_2)/2$;

第五步:以 a 为中心,阈值为 r 的覆盖(即以这两个条件构造 C_j,并找出所有属于 C_j 的样本)。

第六步:如果还有样本未被覆盖,从第一步重复。

1.2 交叉覆盖算法

交叉覆盖和上面的领域覆盖差不多,只是因为领域覆盖只能获得局部最优解,它不具有全局性,为了克服这个缺点,张铃等人后来又提出了交叉覆盖算法,这个算法在解决一些问题时具有很强的优越性。所谓的"交叉覆盖"实际上就是交替覆盖,即先只求出一个领域 C_1,它只覆盖某一类点,而不覆盖其他类的点,然后将被 C_1 覆盖的点删去,对余下的点求另一个领域 C_2,它只覆盖某一类点,而不覆盖其他类点,然后再将被 C_2 覆盖的点删去;如此交叉进行覆盖,直到只剩下最后一类点,将其作为一个领域,这样它的学习与领域覆盖是不同的[3]:

第一步:任取一个第 t 类尚未被覆盖的点 a;

第二步:如果训练集中存在与 a 异类的样本点,计算与 a 最近的异类点到 a 的距离 d_1;不存在异类点,直接转入第三步;

第三步:找出离 a 最远且距离小于 d_1 的同类点 c,两者距离为 d_2;当没有异类点时 d_1 和 d_2 都取最大距离;

第四步:覆盖领域的半径为 $r = (d_1 + d_2)/2$;

第五步:构造以 a 为中心,阈值为 r 的覆盖(即以这两个条件构造 C_j,并找出所有属于 C_j 的样本),并从训练样本中删除被此领域覆盖的样本(a 的信息要保留,以便下次使用);

第六步:如果训练集中还有样本,找到离 a 最近的样本,以它的类型作为 a 的类型,从第二步重复直至训练样本集为空。

1.3 领域覆盖算法及交叉覆盖算法的不足

对于领域覆盖算法,因为每次它都是在局部构造领域,这样就不可避免地出现两个问题,它对输入的训练样本敏感和它的拒识样本可能会很多。对于交叉覆盖算法来说,虽然它在某种程度上具有全局的性质,但是因为交叉覆盖对训练样本敏感,这样通常就造成它的正确率不是太高。从实验数据的表 1 和表 2 的第 2 行、第 3 行与第 4 行的对比中可以清楚地反映两者的缺点,两表的第 2、3 行是交叉覆盖的结果,第 4 行是领域覆盖的结果,一样的训练样本,一样的测试样本,第 2、3 行的拒识样本明显少于第 4 行,但是正确率第 4 行却大于第 2、3 行。

2 集成学习

集成学习是近年来机器学习领域中研究热点之一,它是通过训练多个学习器,再通过一定方式将它们进行组合,从而提高学习系统的泛化能力。集成学习主要有由 2 个步骤组成:个体生成和结论生成。个体学习器学习算法可分为两种:一种是基本的分类、聚类算法,如:文献[10]中,个体学习器使用的就是 k 均值聚类法;另一种是依据学习器组合时的需要而构造特定的学习算法,如:文献[11]中,个体学习器是针对完全随机学习策略的完全随机树。结论生成阶段是将前一阶段得到的学习进行组合的阶段,有两个经典算法:Bagging 和 Boosting。

集成学习的关键是得到较高正确性及差异性的个体学习器[12]。个体学习器的正确性依赖于所选用的个体学习器的学习算法。实验和应用为了使个体学习器间具有差异性,通常使用三种方法:一是使用不同的个体学习器;二是使用不同特征子集;三是使用不同的训练子集[13]。在文中是使用的第一种方法。

3 基于集成学习的覆盖算法

分析领域覆盖和交叉覆盖,可以发现两者的互补性,前者对于一定样本,除了拒识样本之外,会发现它的正确率是很高的,如:表 1 第 1 列第 4 行,算法正确识别了 0.94 * 90 也就是 85 个样本,而拒识样本正好 6 个(因为测试样本取样不是标准的 90 个)。因此实际中只要做好对拒识样本的处理,就让算法性能有很大提升。而对于交叉覆盖发现虽然它的正确率不太高,但它的拒识样本很少。这就给它们两者结合创造了很好的条件,而集成学习恰好提供了一个这样的方法。另一方面交叉覆盖算法有着强的泛化能力,并且交叉覆盖与领域覆盖在覆盖上有很大的差异性,这也是集成学习的关键性条件[14]。综上给出了基于集成学习的覆盖算法(Cover Algorithm Based on Ensemble, CABE)。

CABE 的步骤描述如下:

第一步:从数据集中选择一定样本作为训练样本,分别使用两种交叉覆盖对其进行训练,得到学习器 L_1 和学习器 L_2;

第二步:使用领域覆盖对训练样本进行训练,得到学习器 L_3;

第三步:将第一步取剩的样本作为测试样本,任选其中一个样本 a;

第四步:分别使用 L_1、L_2、L_3 对 a 进行识别,可以识别的则得到对应类别为 T_1,如果不可以则标记 T_2;

第五步:判断 T_3,如果不是特殊标记,则 T_3 作为 a 的类型,进入第七步,否则进入第六步;

第六步:如果 T_1、T_2 都不是特殊标记,以投票法生成 a 的类型;如果 T_1、T_2 中有一个为特殊标记,则 T_3 取领域覆盖算法中离 a 最近的领域的类型,T 取 T_1、T_2 中非特殊标记的那个值,用投票法生成 a 的类型值;如果 T_1、T_2 均为特殊标记,则令 T_1、T_2 取最大领域的类型,而 T_3 取领域覆盖算法中离 a 最近的那个领域的类型,然后对 T_1、T_2、T_3 实行投票算法生成 a 的类型值;

第七步:如果还有没测试的样本,任选一个,转入第四步。

4 实验结果与分析

实验集成过程中,使用了三个学习器,因为训练样本是相同的,为了保证各学习器的差异性,其中交叉覆盖的两个学习器:一个采用重心为中心点,另一个采用随机点为中心点。这个实验有两组,这两组中,第三个学习器是不同的,一个以离重心点最近点作为领域中心的领域覆盖,另一组是以随机产生点的点作为领域中心的领域覆盖。实验使用的数据集是 UCI 的 iris 数据集(属性 4 个,类别 3 个,样本 150 个)。训练和测试样本的取法如下:实验共进行 7 次(对应表中 7 列),第 i 次时(i 从 0 开始),从第 $2i$ 个样本位置开始,5 个一组,前 2 个做训练,后三个做测试,因此训练样本个数为 $60-2i$,测试样本个数为 $90+2i$,表 1 和表 2 中对应列的训练和测试数据是完全一样的,所以两表第 2 行数据完全一样。表中数据取小数点后三位为有效数,并四舍五入,如果小于三位则是准确结果。圆括号内为拒识样本数。

表 3 各算法执行时间对比结果(单位:ms)

CABE(其中的领域算法是以重心为中心)	172	156	125	140
CABE(其中的领域算法是随机产生中心)	141	141	141	156
CCA(中心为重心)	94	93	94	94
CCA(随机中心)	78	94	94	78
CA(中心为重心)	94	78	78	94
CA(随机中心)	94	79	78	78

表 3 中,为了取消内存对不同算法的干扰,这 24 个结果是分 24 次测得,这样就防止因为某些数据已装入内存而对下一次运行时间产生影响。例:如果连续 4 次去测,集成算法(其中的领域算法是以重心为中心)的四次时间为:156,79,31,47。很明显各次之间已经互相影响,由于某些数据已在内存,故时间大减。

观察表 1 和表 2 可以看出,分类正确性有了明显提升,当拒识样本比较多时,算法的改进效果更明显。对比两表,表 1 的提供的结果更理想,也就是当选择以重心作为领域中心的领域覆盖算法在集成时表现的更优秀。另外,表 1 和表 2 的第 2 行数据完全一致是因为交叉覆盖的特点所致,因为训练样本和测试样本一样,并且是以所有训练样本的重点作为领域的中心,所以覆盖的模型就固定了。

对于算法执行时间,各学习器的训练集是相同的,

表 1 集成覆盖算法对比测试结果(CA 以中心为重心)

CABE	0.973(0)	0.967(0)	0.94(0)	0.973(0)	0.933(0)	0.953(0)	0.96(0)
CCA(中心为重心)	0.72(1)	0.873(0)	0.613(1)	0.873(0)	0.68(0)	0.78(7)	0.787(0)
CCA(随机中心)	0.74(1)	0.827(0)	0.893(1)	0.947(0)	0.733(2)	0.913(1)	0.793(0)
CA(中心为重心)	0.94(6)	0.9(12)	0.787(24)	0.947(6)	0.887(9)	0.92(6)	0.907(12)

表 2 集成覆盖算法对比测试结果(CA 以随机点为重心)

CABE	0.73(0)	0.96(0)	0.893 (0)	0.973(0)	0.893(0)	0.973(0)	0.967(0)
CCA(中心为重心)	0.72(1)	0.873(0)	0.613(1)	0.873(0)	0.68(0)	0.78(7)	0.787(0)
CCA(随机中心)	0.92(0)	0.84(0)	0.633(1)	0.82(0)	0.7(0)	0.713(2)	0.973(0)
CA(随机中心)	0.913(10)	0.913(9)	0.887(2)	0.953(5)	0.86(13)	0.913(10)	0.913(9)

注:上面两表中第一行表示集成之后的测试结果,后面三行分别是此集成器用到的三个分类器对应的结果。

因此读入内存的数据可以被几个学习器所共用，这样用于读取数据部分的时间增加的并不会太大，并且由于此集成算法是类似投票算法的简单算法，所以运行时间增加也是可以接受的，而且不像大多识别拒识样本算法那样花费大量时间在判断样本领域上（因为本算法每个样本只判断三个领域）。

从表 3 也可以看出，总起来说三个个体学习器的集成算法中，个体学习器学习加上集成总的执行时间不超过个体学习器时间的二倍。从上面的实验结果及分析可以看出，集成算法无论在效果还是在执行时间上都是可行有效的。

5　结束语

基于集成学习的覆盖算法是通过对领域覆盖算法中拒识样本的处理，仅使用三个学习器就可以使算法性能有了很好的提升，它提供了一种增加领域覆盖算法泛化能力的方法。集成学习和覆盖算法结合是一种新的尝试，今后可进一步研究覆盖算法的改进及其与集成学习的结合。

参考文献：

[1] 张　铃，张　钹. M－P 神经元的几何意义及其应用[J]. 软件学报，1998，9(5)：334－338.

[2] 张　铃，张　钹. 前向神经网络设计问题的回顾和探索[J]. 计算机工程和科学，1998，20(4)：1－10.

[3] 张　铃，张　钹. 多层前向网络的交叉覆盖设计算法[J]. 软件学报，1999，10(7)：337－342.

[4] 吴　涛，张　铃. 机器学习中的核覆盖算法[J]. 计算机学报，2005，28(8)：1295－1301.

[5] 张燕平，张　铃. 机器学习中的多侧面递进算法 MIDA[J]. 电子学报，2005，33(2)：327－331.

[6] 张　铃，吴　涛. 覆盖算法的概率模型[J]. 软件学报，2007，18(11)：2691－2699.

[7] 王倩倩，段　震，张燕平. 基于交叉覆盖算法的文本分类[J]. 计算机技术与发展，2007，17(6)：113－115.

[8] 汪小寒，陈　洁. 基于核覆盖算法的煤价预测[J]. 计算机技术与发展，2006，16(12)：81－85.

[9] 张燕平，张　铃. 构造性核覆盖算法在图像识别中的应用[J]. 中国图像图形学报，2004，14(9)：1304－1308.

[10] 唐　伟，周志华. 基于 Bagging 的选择性聚类集成[J]. 软件学报，2005，16(4)：496－502.

[11] 俞　扬，周志华. 集成学习中完全随机学习策略研究[J]. 计算机工程，2006，32(17)：100－102.

[12] 梁英毅. 集成学习综述[EB/OL]. 2006. http://soft.cs.tsinghua.edu.cn/keltin/docs/ensemble.pdf.

[13] Garcia－Pedrajas N. Nonlinear Boosting Projections for Ensemble Construction[J]. Journal of Machine Learning Research，2007(8)：1－33.

[14] 周志华. 通过集成学习进行知识获取[J]. 重庆邮电大学学报，2008，20(3)：361－362.

（上接第 75 页）

```
    bm->UnlockBits(&data);
    return true;
}
```

经过验证，一幅 800 * 600 的 24 色位图，采用此方法的灰度化过程耗时 0.046875 秒，处理速度非常快，可以满足大部分情况下的实际图像处理需要。

3　结束语

介绍了三种基于 GDI＋的图像灰度化实现方法，进行了性能、实现复杂度等方面的比较。结论是直接读写内存图像数据法处理速度最快，但编码稍微繁琐；利用 GDI＋的色彩变换方法处理速度快，但耗时比读写内存图像数据法多大约 1 倍，编码简明；直接读写像素法处理速度非常慢，不能满足实际数字图像处理需要。

介绍的基于 GDI＋的图像灰度化实现方法和比较结论，可为开发人员的实际开发应用提供借鉴和参考。代码在 Visual Studio C＋＋.net 2005，Windows Xp 环境下调试编译通过。

参考文献：

[1] Pratt W K. Digital Image Processing[M]. [s.l.]: John Wiley & Sons，1991.

[2] Gonzalez R C，Woods R E. Digital Image Processing[M]. [s.l.]: Prentice Hall Press，2007.

[3] 彭召意. 16 位位图的灰度化处理方法[J]. 中国包装工业，2002(6)：150－151.

[4] Microsoft Corporation. Microsoft Developer Network(MSDN)[EB/OL]. 2008. http://www.Microsoft.com/msdn.

[5] 闫宇晗，常　鑫. 在 C# 中用 GDI＋实现图形动态显示[J]. 计算机技术与发展，2006，16(12)：117－118.

[6] 黄烟波，赵旭华，刘中宇. 基于.NET 的流程图绘制程序的设计与实现[J]. 计算机技术与发展，2007，17(5)：231－233.

[7] 周鸣杨，赵景亮. 精通 GDI＋编程[M]. 北京：清华大学出版社，2004.

[8] 周长发. 精通 Visual C＋＋图像处理编程[M]. 北京：电子工业出版社，2006.