

一种改进的贪婪式覆盖算法

宋杰,程家兴,许中卫,周瑛

(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)

摘要:文中对覆盖算法进行了介绍和分析,提出了一种基于贪婪算法思想的改进的覆盖算法,称之为贪婪覆盖算法。通过对覆盖初始中心选择方式的改进,减少覆盖数量。通过实验,对比目前已有的几种实现方法,覆盖数量有了较大的下降,明显提高了分类识别的速度。

关键词:覆盖算法;神经网络;贪婪算法

中图分类号:TP183

文献标识码:A

文章编号:1673-629X(2006)08-0113-03

An Improved Greedy Covering Algorithm

SONG Jie, CHENG Jia-xing, XU Zhong-wei, ZHOU Ying

(Ministry of Education Key Lab. of Intelligent Computing and Signal Processing, Anhui University, Hefei 230039, China)

Abstract: By the analysis of covering algorithm, this paper gives an improved covering algorithm inspired by greedy algorithm. The improvement is implemented by the change of selecting covering start point. Comparing with several realized method, this method decreased the number of covers and improved recognized speed by experiments.

Key words: covering algorithm; neural network; greedy algorithm

0 引言

覆盖算法是一种基于 M-P 神经元的构造型神经网络算法,自提出以来已经得到越来越广泛的应用,目前已成功用于文字识别、图像处理和雷达信号识别等领域。其基于几何意义的构造方法便于理解,处理方便。文中提出了一种改进的覆盖算法,使网络复杂度下降且识别速度有了进一步的提高。

1 覆盖算法原理

张铃等人提出了覆盖算法^[1],为分类学习提供了一个新的方法。该算法具有容易理解、识别率高、计算速度快、可保证训练集中的样本 100% 的识别率的优点。张还陆续证明了该神经网络可以等价于一个三层前馈神经网络模型以及三层神经网络和基于核函数支持向量机(SVM)的内在关系^[2]。覆盖算法将神经网络的最优设计问题转化成某种求最优覆盖的问题。在应用上,它使得基于原理处理海量样本集成为可能,被认为是对 SVM 传统机器学习理论的重要贡献。

1.1 覆盖算法的基本概念和性质

M-P 神经元是一个 n 输入、单输出的元件,其输入

与输出的关系为 $y = \text{Sgn}(\sum w_i x_i - a)$ 。若令 $\sum w_i x_i - a = 0$,则此式表示为一个超平面方程。于是从几何上可将神经元看成是一个空间分类器,即落在正半空间的点对应的输出为 1;落在负半空间上的点对应的输出为 -1。以上是神经元的超平面几何模型,这种模型对理解单个神经元是很有帮助的,但它对理解整个神经网络的帮助不大,因为众多超平面之交在空间构成的划分是非常复杂的。

根据神经元几何模型,提出神经网络覆盖算法^[3]。设样本空间中的样本是分布在 $n+1$ 维空间中某个中心在原点的球面 S^n 上,若不然,可通过变换: $T: D \rightarrow S^n$, $T(x) = (x, \sqrt{R^2 - \|x\|^2})$ 将样本点映射到球面 S^n 上,其中, $R \geq \max\{\|x\|\}$ 。与球面相交的超平面 $\langle \omega, x \rangle - \theta = 0$,将球面分成: $H^+ : \langle \omega, x \rangle - \theta > 0$ 和 $H^- : \langle \omega, x \rangle - \theta < 0$ 两部分,其中 $\langle \omega, x \rangle$ 是内积。称球面上位于超平面 $\langle \omega, x \rangle - \theta = 0$ 所分割的正半空间的部分为球面上的“球形领域”,若 ω 与 x 等长,则原点就是这个球形领域的中心。如图 1 所示。以每个球形领域作为一个神经元,定义其功能函数 $f(x)$ 和判别规则,即可训练分类器。这种基于覆盖的分类器可适用于多类别情形,简单高效,且对训练样本 100% 可识别,这是一般的分类器难以做到的。

1.2 覆盖算法分析

设学习样本共有 N 类,记为: $X = \{X_1, X_2, \dots, X_N\}$,学习过程中构造第 k 类学习样本 X_k 的“球形领域”的方法是:任取其中尚未覆盖的点 a_i ,按公式

$$d^1(\omega) = \max_{x \in X_k} \langle \omega, x \rangle \quad (1)$$

收稿日期:2005-12-03

基金项目:国家自然科学基金中外合作特别基金资助项目(60111120622)

作者简介:宋杰(1966-),男,安徽合肥人,副教授,博士,研究方向为智能计算、模式识别、生物信息学。

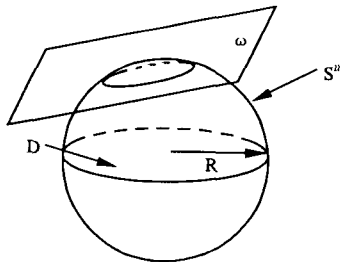


图 1 超球面投影示意图

$$d^2(\omega) = \min_{x \in X_i} \{ \langle a_i, x \rangle - \langle a_i, x \rangle \} d^1(\omega) \quad (2)$$

$$d(\omega) = \frac{1}{2} (d^1(\omega) + d^2(\omega)) \quad (3)$$

计算, 作以 a_i 为中心、阈值 $\theta = d(\omega)$ 的覆盖 $C(a_i)$: $\langle \omega, x \rangle - \theta > 0$, 尽可能覆盖更多的样本点, 并按此方法求出样本的全部覆盖。取功能函数

$$F_1: y = \sigma(\langle \omega, x \rangle - \theta), \sigma(x) = \begin{cases} x, & x > 0 \\ 0, & \text{其它} \end{cases} \quad (4)$$

识别的方法是: 给定一个样本, 若它属于某类覆盖的一个“球形领域”, 即可确定其类别, 否则, 若它不属于任何类别覆盖的一个“球形领域”, 则拒识, 或按 $\min \{d(x, C(a_i))\}$ 确定其类别归属。

2 覆盖算法神经网络构造

2.1 网络结构

神经网络的覆盖算法是把求解样本集 S 的 N 类分类问题转化成在样本空间构造覆盖簇 $\{C_i\}$, 使每个覆盖 C_i 只盖住同一类点且 C_i 并覆盖整个 $\{x_1, \dots, x_n\}$ 。设已求得覆盖组 C_1, \dots, C_K , 取三层神经网络, 隐层取 K 个神经元, 每个神经元为一个覆盖。输出层取 N 个神经元, 第 i 个神经元的输入为覆盖第 i 类点的覆盖的输出, 其激励函数为竞争函数。这样的三层网络, 就可对 S 进行分类^[4]。

2.2 构造分类

按照覆盖算法的基本原理, 具体有两类不同的实现方法。

一类是每个覆盖不包括异类, 这种方法每个覆盖相互独立, 具有平等的关系 (又称一般覆盖)。

设每次覆盖可用区域 C_{ij} 表示, 其中, i 表示覆盖的序号, j 表示该覆盖所属的类别。则本类覆盖第 k 类的区域可用 $\sum_{i=1}^K C_{ij} (j \in k)$ 表示。

它的优点是错误率低。缺点是没有利用先前覆盖点的信息, 覆盖数量多, 且拒识率非常高。

另一类是具有先后顺序的覆盖, 每个覆盖的优先权不同, 前面的覆盖比后面的覆盖的优先级高, 如果一个样本落入两个以上的覆盖, 则归入前面的覆盖。

对于这种覆盖, 第 i 次覆盖所得到的实际的区域是图 2 中的阴影部分:

$$A_i = C_i - \sum_{x=1}^{i-1} C_x \quad (5)$$

因为前面的覆盖 (中的样本) 需减去 (所以又称挖点覆

盖), 如图 2 所示。

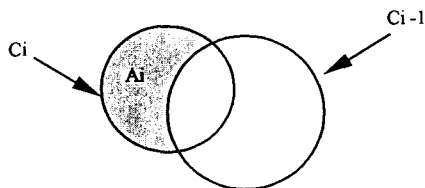


图 2 逐次覆盖示意图

这种方法在使用时要注意的在识别时要按先覆盖先识别的顺序。

文中基于减少神经元数量的目的采用后一种方法。

初步的改进方法基本同上, 但在一个覆盖结束后并不立即认定该覆盖, 而是用已覆盖的点计算中心, 并以此为中心再次重新覆盖, 如覆盖比上次样本多, 则用此覆盖替代, 直到覆盖数量不再增加。

这种方法的弱点是最终覆盖的质量受先验知识和初始种子样本的选择以及样本分布的影响较大, 因此更进一步提出了贪婪覆盖算法。

3 贪婪覆盖算法

笔者针对初始点的选择弱点加以改进。文中的方法受到贪婪算法的启发, 称之为贪婪覆盖算法。在经过几次覆盖后, 如何选取下一个覆盖的中心, 前述方法都是以最后碰到的异类点为中心。这样的方法容易陷入一些较小的局部中。如图 3 以一个简单的二维两类分类为例, 说明并不是最近异类点是最佳的下次覆盖中心。图中, 黑白点分别代表不同的样本, 圆周上的数字是覆盖的序号。左图是一个按最后所遇到的异类点为下一个覆盖的起始中心点, 右图为贪婪覆盖算法择优选择覆盖中心的结果。其中左图用了 6 个覆盖, 而右图只用了 3 个。

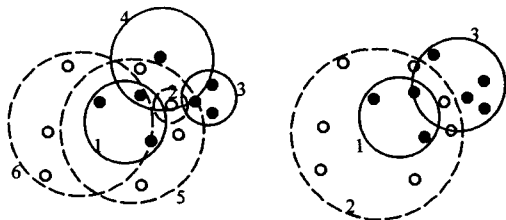


图 3 不同的覆盖方法顺序和数量比较

贪婪算法的思想是在最后一个异类点和已有的各覆盖中心分别为初始覆盖中心求覆盖, 并选择其中一个覆盖样本数最多的为实际覆盖。

贪婪覆盖的算法是:

- 1) 求第一个覆盖。
- 2) 先以遇到的最后一个异类样本点为中心求新的覆盖。
- 3) 再依次从已有的覆盖中心构建新的覆盖, 如新覆盖比前面的覆盖点数量多, 则用新覆盖代替旧覆盖。
- 4) 从 2、3 步中选出最好的一个覆盖保存此覆盖的相关数据 (覆盖中心, 半径, 类型等)。
- 5) 在数据集中删除选中覆盖包含的点。

6)返回 2,反复执行,直到所有点都覆盖完毕。

4 实验与分析

为评估本算法的性能,使用了标准的 UCI 测试数据集进行测试。为使结果和其它文献具有可比性,选用了 iris, glass, liver 几个数据集。

表 1 数据集参数

| 数据集 | 样本数 | 特征数 | 类别数 |
|-------|------|-----|-----|
| Iris | 150 | 4 | 4 |
| Glass | 214 | 10 | 7 |
| Liver | 345 | 6 | 2 |
| car | 1728 | 6 | 4 |

实验分为两个方面,一是和其它文献使用的覆盖算法进行比较,二是对不同的覆盖算法进行自测。

首先,按文献[5]所采用的测试条件,即用 10 交叉法测试。用 MATLAB6.5 编程,测试结果和文献比较见表 1。

表 2 算法正确率比较 (%)

| 数据集 | 文献[5] | 文献[6] | 贪婪覆盖 |
|-------|-------|-------|-------|
| Iris | 71.54 | 96.68 | 96.81 |
| Glass | / | 91.19 | 96.05 |
| Liver | / | 93.04 | 92.89 |
| car | 80.75 | / | 82.11 |

同时,使用不同的覆盖算法采用留一法进行自测,测试结果(平均覆盖数已取整数)列于表 3。

表 3 不同的覆盖方法所用的覆盖数比较

| 数据集 | 一般覆盖 | | 挖点覆盖 | | 贪婪覆盖 | |
|-------|------|-------|------|-------|------|-------|
| | 覆盖数 | 正确率% | 覆盖数 | 正确率% | 覆盖数 | 正确率% |
| Iris | 29 | 94.64 | 12 | 95.33 | 9 | 95.23 |
| Glass | 22 | 94.33 | 7 | 96.73 | 6 | 97.20 |
| Liver | 143 | 90.2 | 121 | 91.4 | 65 | 91.4 |
| car | 458 | 91.4 | 228 | 93.6 | 145 | 93.5 |

由表 3 可见,错误率基本不变,但覆盖数量有了明显的下降。由于覆盖的数量对应着隐层神经元的数量,这意味着在对测试样本进行识别时,由于网络神经元数量的减少而提高了识别速度,减少了识别时间。

对于求得第 i 次最优覆盖,易知其所需使用的普通覆盖次数是 i 次。所以若需覆盖数量 n ,整个覆盖过程需要的次数为 $n(n+1)/2$ 次。但由于贪婪覆盖算法的覆盖点数少,即 n 值大幅减小,则覆盖算法的覆盖数在样本数量

多时,远小于一般覆盖。同时,为防止覆盖数量多时时间过长,也采取了一些措施,随机从前面的覆盖中限定选若干个。笔者在实验中取 10 个,与不限制相比,在覆盖数量小于 10 时,不起作用,在覆盖数量多时,保证计算时间限定在一个数量级内。根据贪婪算法的特点,通过几个大覆盖数样本的实验,是否用此限制,结果互有优劣,但相差不多。

文献[6]提出的核偏移算法,虽然它能获得局部最优,但显然,局部最优不见得会导致全局最优,实验结果证明了这一点。但同时,它所耗费的时间代价是巨大的,与一般覆盖算法相比,其计算时间呈指数增长。在目前没有哪种方法被证明最优的情况下,贪婪覆盖算法是一种求解的有效途径。

5 结 论

本算法的特点是以增加少量训练的时间而大量减少识别时所用的时间,但增加的时间是可以控制的。处理方法简单易懂,效果明显。并且由于覆盖算法的非线性特点,可以构造非常复杂的划分边界。在实际的应用中,有相当多的应用对学习时间不是特别敏感。这些应用就可以采用贪婪覆盖算法。与目前流行的 SVM 相比,它也有自己的优势,SVM 对于两类问题,有较优的解,但对多类问题则需转换成多个两类问题,计算时间长。而覆盖算法可直接处理多类识别问题,并且相对时间较短。两种方法各有所长,可根据使用对象选择。

参考文献:

[1] 张 铃. A Geometrical Representation of McCulloch Pitts Neural Model and Its Applications[J]. IEEE Trans on Neural Networks,1999,10(4):925-929.
[2] 张 铃. 基于核函数的 SVM 机与三层前向网络的关系[J]. 计算机学报,2002,25(7):697-700.
[3] 张 铃,张 钺. M-P 神经元模型的几何意义及其应用[J]. 软件学报,1998,9(5):334-338.
[4] 张 铃,张 钺,殷海风. 多层前向网络的交叉覆盖算法[J]. 软件学报,1999,10(7):737-742.
[5] 毛军军,吴 涛,郑婷婷. 基于商空间的构造性分层竞争网络算法[J]. 微机发展,2005,15(4):37-39.
[6] 赵 姝,张燕平,张 媛,等. 基于交叉覆盖算法的改进算法——核平移覆盖算法[J]. 微机发展,2004,14(11):1-3.

(上接第 112 页)
域的核心东西,慎重选择。

参考文献:

[1] Geoff C C,Keeton F B. JAVA 完全探索(第 2 版)[M]. 师夷工作室译. 北京:中国青年出版社,2001.
[2] 尉哲明,郝建文. JAVA 中利用内部类简化程序的编写[J].

微机发展,2003,13(3):41-44.
[3] Wampler B E. JAVA 与 UML 面向对象程序设计[M]. 王海鹏译. 北京:人民邮电出版社,2002.
[4] 张海藩. 软件工程导论[M]. 北京:清华大学出版社,2003.
[5] 王克宏,董 丽. Java 技术及其应用[M]. 北京:高等教育出版社,1999.