第16卷 第2期 自 动 化 学 报 Vol. 16, No. 2
1990 年 3 月 ACTA AUTOMATICA SINICA Mar., 1990

快速非参数分类器 BDPATCH-GBB

李 先 愍 王 庆 人
(南 开 大 学)

摘 要

本文提出了一种快速的浓缩近邻分类器 BDPATCH-GBB，并从压缩样本集和减少计算量两方面着手提高分类器的性能. 实验表明，与其它近邻分类器相比，BDPATCH-GBB 是高效的.

关键词——近邻分类，分支定界，样本分割，距离计算.

一、引 言

K-近邻（K-NN）方法被证明在分类中是一种有效的非参数技术. 然而，当训练集样本容量很大时，其占用的内存以及对一个新模式进行分类时所消耗的机时也非常大. 为了克服这一严重缺陷，可以采用在保持 NN 规则优点同时压缩样本集的大小，或是通过适当地组织样本集来减少搜索中的计算量等方法. 样本集的压缩是通过对样本模式的编辑[1]、浓缩[2]和补缀[3]等算法来实现的，而计算时间的压缩则是通过采用适当的数据结构和引入某些规则[4,5]以去掉一些不必要的计算来完成的. 本文提出了一种快速的浓缩近邻分类器 BDPATCH-GBB.

二、BDPATCH-GBB 过程

1. 编辑与浓缩

在分类过程中，不同类别的样本模式交叉分布会严重影响分类器的识别率. 为了除去那些在类与类交界处相互交叉的样本，我们首先对样本集进行编辑. 为减少距离计算，在编辑基础上进一步对样本集进行浓缩. 所用算法见文献[1,3]. 经过编辑和浓缩两过程后，得到了我们使用的浓缩样本集 S.

2. 广义分支定界算法

设 $S=\{X_1, X_2, \cdots, X_N\}$. 现在需要寻找测试样本 X 在 S 中的最近邻. 为进一步提高分类器的速度，我们引进了广义分支定界搜索算法. 为此，需要先根据 S 形成一棵搜索树. 在这里，我们用 2-means 聚类算法来形成搜索树. 树的每一结点 n 代表样本的一个类，结点参数如下：S_n 为结点 n 上的样本类；M_n 为结点 n 上的样本均值；N_n 为结点 n 上

本文于 1987 年 8 月 11 日收到.

的样本个数；$r_n = \max\limits_{X \in S_n}\{d(X, M_n)\}$ 表示从 M_n 到 X 的最远距离。$d(\cdot, \cdot)$ 是本文所用的距离度量。在根结点上，$S_n = S$，$N_n = N$。

由 S 形成搜索树后，对每个结点 n 进行测试，确定 X 的最近邻是否属于 S_n。在搜索过程中，沿用 FN 算法[4]中的剪枝规则。

规则 1. 如果 $B + r_n < d(X, M_n)$，则 $X_i \in S_n$ 不可能是 X 的最近邻。其中 B 是到目前为止所找到的 X 的最近邻到 X 的距离。

规则 2. 如果 $B + d(X_i, M_n) < d(X, M_n)$，则 X_i 不可能是 X 的最近邻，$X_i \in S_n$。

规则 1 是在搜索非终结结点时使用的，而规则 2 则是在搜索达到终结结点时使用．我们的广义分支定界算法如下：

```
PROCEDURE GBB
    1    B ← +∞;
    2    Set OPEN LIST to empty;
    3    CURRENT NODE ← Root;
    4    Repeat
    5    If CURRENT NODE is not a terminal Then
    6    expand CURRENT NODE, and
           put the offsprings
    7    of CURRENT NODE into OPEN LIST;
    8    Else
    9    test for Rule 2;
   10    test for Rule 1, delete some nodes in OPEN LIST;
   11    choose the nearest node;
   12    Until OPEN LIST is empty;
END GBB
```

上述算法中，OPEN 表用以存放没有被考虑的树结点。第 1—3 步是初始化。第 9 步，当当前结点是终结结点时，测试规则 2 这时可能修改 B 的值。第 11 步用以在 OPEN 表中寻找和 X 距离最近的结点 n，并把 n 置成当前结点。

三、实 验 结 果

我们在 Super AT 微机上对 BDPATCH-GBB，FN 算法及 BDPATCH[3] 进行了仿真试验。训练样本和测试样本独立地由下述分布函数生成：

$$x = \sigma_1(-2\ln x_{2n})^{\frac{1}{2}} \cdot \cos 2\pi x_{2n+1} + \mu_1,$$

$$y = \sigma_2(-2\ln x_{2n})^{\frac{1}{2}} \cdot \sin 2\pi x_{2n+1} + \mu_2,$$

$$z = \sigma_3(-2\ln x_{2n})^{\frac{1}{2}} \cdot \cos 2\pi x_{2n+1} + \mu_3,$$

其中，$\{x_n\}$ 是由随机数发生器产生的一个随机实数序列，每个 $x_n \in [0,1]$。

实验是在两类问题上进行的。两类数据通过选择两组参数 $\{\sigma_i, \mu_i\}$ 和 $\{\sigma_i', \mu_i'\}$，$i = \overline{1,3}$ 而产生。在实验中共生成 15 个训练集，其中每三个为一组，容量相同。这样，

对于一个容量值可以把上述三种算法比较三次。另外生成的 512 个测试样本用来估计各自的识别率及分类速度。三种分类器在同一训练集下的平均性能列于表 1。图 1 反应了三种分类器在不同的样本容量下对测试样本的分类时间。

表 1　三种分类器平均性能列表

原始训练集样本数	浓缩集 S 样本数	FN 算法		BDPATCH		BDPATCH-GBB		
		时间	误识率%	时间	误识率%	时间	误识率%	距离计算次数
64	17	27″	7.61	15″	6.38	14″	6.38	13
128	25	39″	6.9	21″	5.66	17″	5.66	16
256	42	57″	7.1	35″	5.86	23″	5.86	22
512	57	121″	6.12	47″	4.42	29″	4.42	27
1024	86	206″	7.59	111″	4.88	37″	4.88	34

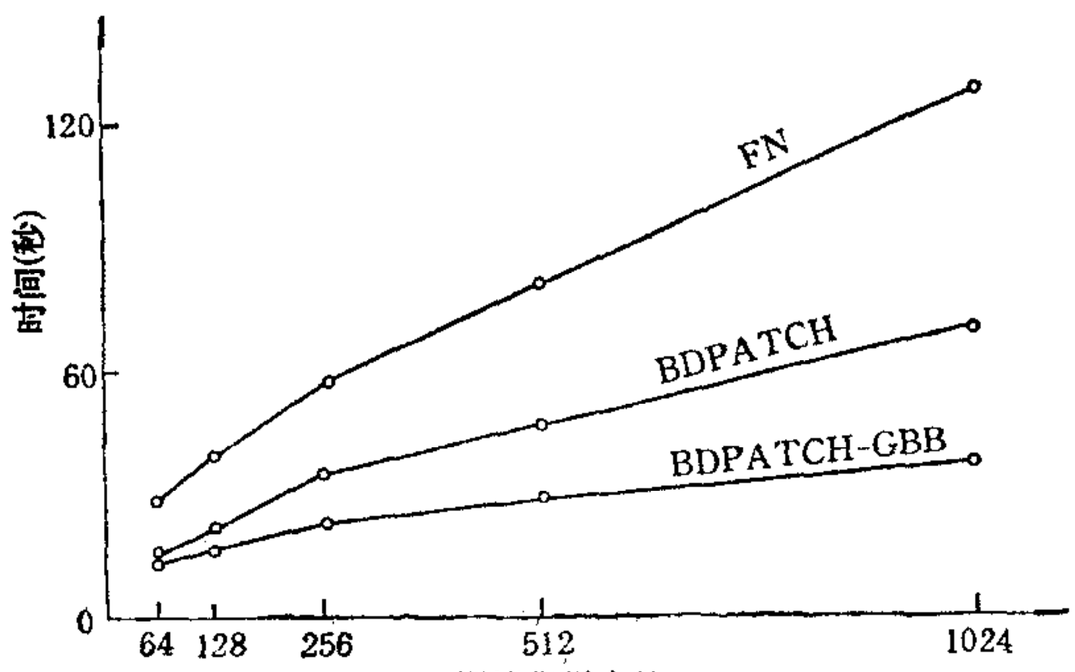

图 1　三种分类器时间增长示意图

四、结　论

观察和分析上述试验结果可以得到如下结论：

(1) 和 FN 算法相比较，BDPATCH-GBB 在分类速度和识别率上均有明显的提高。当训练样本集很大时，BDPATCH-GBB 不会象 FN 算法那样由于占用内存庞大而带来实现上的困难。

(2) 与 BDPATCH 分类算法相比，BDPATCH-GBB 保持了它的识别率，但在分类速度上却有较好的提高。为了便于比较，表 1 的最后一列给出了 BDPATCH-GBB 分类新模式 X 的距离计算次数，而 BDPATCH 分类 X 的距离计算次数则由表中第二列给出。由表可见，随着 S 中样本模式的增加，BDPATCH-GBB 的距离计算次数也相应增加，但其增长速度远远没有 BDPATCH 的快。可以看出，训练集样本容量越大，分类速度改善就越明显，当训练集样本容量为 1024 时，BDPATCH-GBB 的分类速度比 BDPATCH 约快 1 倍。

参 考 文 献

[1] Devijver, P. A. and J. Kittler, On the Edited Nearest Neighbor Rule, Proc. 5th Int. Conf. Patt. Recogn., 72—80, Miami FL, 1980.

[2] Hart, P. E., The Condensed Nearest Neighbor Rule, *IEEE Trans. Inform. Theory* IT-14(1968), 515—516.

[3] 王庆人，补级式浓缩近邻分类器 BDPATCH，自动化学报，14(1988)，106—111.

[4] Fukunaga, K. and P. M. Narendra, A Branch and Bound Algorithm for Computing k-nearest Neighbors, *IEEE Trans.*, July, 1975, 750—753.

[5] Enrique Vidal Ruiz, An Algorithm for Finding Nearest Neighbors in (Approximately) Constant Average *Time, Patt. Recogn. Lett.*, 4(1986) 145—157.

FAST NONPARAMETRIC CLASSIFIER BDPATCH-GBB

Li Xianjing Wang Qingren

(*Nankai University*)

Abstract

A fast CNN classifier BDPATCH-GBB is presented. Measures are taken to reduce the sample size and the amount of computation so as to enhance its performance. Experiments show that it is more effective when compared with other NN classifiers.

Key words ——Nearest neighbour classification; branch-and-bound algorithm; sample decomposition; distance computation.