第32卷 第16期 计 算 机 工 程 2006年8月
Vol.32 № 16 Computer Engineering August 2006
·软件技术与数据库· 文章编号：1000—3428(2006)16—0055—03 文献标识码：A 中图分类号：TP311

SPRINT 算法的改进

刘友军，汪林林
（重庆邮电学院经济管理学院，重庆 400065）

摘 要：引出了纯区间的概念后，提出了一种基于纯区间归约的数值型属性处理方法对 SPRINT 算法进行改进。该方法将属性值域用等宽直方图的方法划分为多个区间，对纯区间进行归约，对非纯区间进行精确计算，保证了分裂精度，减小了计算量。
关键词：决策树；SPRINT 算法；纯区间归约；Gini 指数

Improvement of SPRINT Algorithm

LIU Youjun, WANG Linlin
(School of Economics Management, Chongqing University of Posts and Telecommunications, Chongqing 400065)

【Abstract】 This paper introduces the concept of pure interval, proposes a new splitting method based on pure intervals reduction to deal with numeric attributes for SPRINT algorithm. The method divides the numeric attributes to many intervals with equal-width histogram, reduces the pure intervals, calculates exactly the minimum gini value in the impure intervals, ensures the accuracy of split result and reduces computation.
【Key words】 Decision tree; SPRINT algorithm; Pure intervals reduction; Gini index

数据分类是数据挖掘中的一种重要的分析方法，通过学习训练集构造一个分类函数或分类模型，应用该函数或模型将数据记录进行分类。常见的分类模型的构造方法有统计方法（如贝叶斯方法）、机器学习方法（如决策树方法）、神经网络方法、粗糙集方法和遗传算法等。相对其它的分类算法，决策树的计算量较小、容易转化成分类规则，挖掘出的分类规则准确性较高，便于理解，因此，决策树在数据分类方面有着广泛的应用。

从Quinlan提出了ID3算法以后，相继出现了C4.5、SLIQ、SPRINT、PUBLIC等决策树分类算法。面对海量数据集上的数据挖掘，决策树分类算法在可伸缩性方面的主要问题是训练集受内存容量的限制。为了改进算法的可伸缩性，SPRINT算法定义了新的数据结构，消除了所有内存限制，运行速度较快，而且能很好地支持并行处理。但是，在处理数值型属性方面，SPRINT 算法采用了精确查找技术，这种技术要对整个训练集进行预排序，工作量很大。对于超大数据集，当属性含有大量的不同取值时，效率非常低。我们的算法主要从处理数值型属性方面对 SPRINT 算法进行改进。

1 SPRINT 算法

1.1 分裂指数

分裂指数是用来度量属性分裂规则优劣程度的一个量度。gini 指数（gini index）是一种能够有效地搜索极佳分裂点的分裂指数，在 SPRINT 算法中就是采用了 gini 指数。

设一个数据集 S 有 n 条记录，它们分属于 c 个互不相关的类，则集合 S 的 gini 值是

$$gini(S)=1-\sum_{j=1}^{c}p_j^2 \tag{1}$$

其中$p_j=m/n$，m为S中属于类j的记录数。

如果使用分裂规则$cond$将S划分为S_1和S_2两个子集，则该规则的度量值记为$gini^D(S,cond)$，定义如式(2)所示：

$$gini^D(S,cond)=\frac{n_1}{n}gini(S_1)+\frac{n_2}{n}gini(S_2) \tag{2}$$

其中n_1、n_2分别为S_1、S_2的记录数。

$gini^D(S, cond)$越小，表明分裂规则越好。

1.2 SPRINT 基本思想

构造决策树的基本策略是采用贪心方法，用自上而下的递归方式生成树。生成一棵决策树一般分为创建阶段和剪枝阶段。SPRINT 算法也不例外。

创建树算法如下：

输入：训练集样本 T

输出：一棵决策树

步骤：

(1)如果 T 满足停止扩展的条件，则返回；

(2)对于每一个属性A_i，找到A_i的一个值或值集V_i，它将产生以A_i为测试属性的最佳分裂；

(3)比较各个属性的最佳分裂，选择一个最佳的将T分为T_1和T_2；

(4)递归地对T_1和T_2生成决策树。

算法中，集合T、T_1、T_2分别代表树中的结点，其中T_1和T_2是T的两个分支结点。最后生成的决策树是一棵二叉树。

SPRINT 算法的剪枝采用了最小描述长度（Minimum Description Length，MDL）原则。由于篇幅的关系，相关的详细内容参看文献[3]。

1.3 数值型属性的分裂

对于一个数值型属性A，它的分裂形式为$A\leq v$。因此，可以先对数值型属性排序，假设排序后的结果是$v_1, v_2, ..., v_n$，因为分裂只会发生在两个结点之间，所以有n-1 种可能性。通

作者简介：刘友军(1976—)，男，助教、硕士生，主研方向：数据挖掘；汪林林，教授
收稿日期：2005-11-29 **E-mail：**liuyj@cqupt.edu.cn

常取中点$(v_i+v_{i+1})/2$作为分裂点。从小到大依次取不同的分裂点，取gini值最小的点作为最佳分裂点。

这种寻找最佳分裂点的方法能够找到最精确的分裂点。但是对于数值型属性，首先要对整个训练集进行预排序，然后将每两个结点之间的中点值都作为分裂点来计算 gini 值，工作量很大，特别是对于超大数据集，当属性含有大量的不同取值时，效率非常低。

2 SPRINT 算法的改进

针对精确查找技术的工作量大的缺点，我们提出了一种纯区间归约的方法，将训练集划分为多个区间，对纯区间进行归约，对非纯区间进行精确查找，减小了计算的工作量。

2.1 纯区间归约法的原理

定义 1 对于某个数值型属性，如果在区间$[v_b, v_t]$中的所有记录都属于同一个类C_i，则称该区间为C_i的纯区间。

定理 1 设$f(x)$在$[a, b]$上连续，在(a , b)内具有一阶和二阶导数，那么

(1)若在(a , b)内，$f''(x)>0$，则$f(x)$在$[a , b]$上的图形是凹的；

(2)若在(a , b)内，$f''(x)<0$，则$f(x)$在$[a , b]$上的图形是凸的。

设待划分的数据集为S，$[v_l, v_u]$为一个区间，其中：

n：数据集 S 的大小；

c：S 中类的个数；

x_i：类i中小于或等于v_l的记录数；

y_i：类i中小于或等于v_u的记录数；

c_i：类i的所有记录数；

n_l：小于等于v_l的记录数（即为$\sum_{i=1}^{c} x_i$）；

n_u：小于等于v_u的记录数（即为$\sum_{i=1}^{c} y_i$）；

由式(2)得$gini^D$在点v_l的值如下：

$$gini^D(S, a \le v_l) = \frac{n_l}{n}(1-\sum_{i=1}^{c}(\frac{x_i}{n_l})^2) + \frac{n-n_l}{n}(1-\sum_{i=1}^{c}(\frac{c_i-x_i}{n-n_l})^2) \quad (3)$$

对类C_k的一个纯区间$[v_b, v_t]$，在式(3)中，x_i（i=1,2,…,c）中表示该纯区间第k个类的记录数为x_k ，只有x_k变化，令$x_k=x$，$c_k=C$，则式(3)可以转化为一个关于x的函数。对于式(3)，令：

$$n_l = \sum_{i=1}^{c} x_i = A + x \quad , \quad \sum_{i=1}^{c} x_i^2 = B + x^2 \quad ,$$

$$\sum_{i=1}^{c}(c_i - x_i)^2 = D + (C - x)^2$$

其中 A、B、C、D 均为大于等于 0 的常数，则式(3)转化为关于 x 的函数，如式(4)所示：

$$f(x) = 1 - \frac{B + x^2}{n(A + x)} - \frac{D + (C - x)^2}{n(n - A - x)} \quad (4)$$

式(4)的一阶导数为

$$f'(x) = \frac{1}{n}(\frac{B + x^2}{(A + x)^2} - \frac{2x}{A + x} - \frac{D + (C - x)^2}{(n - A - x)^2} + \frac{2(C - x)}{(n - A - x)}) \quad (5)$$

式(4)的二阶导数为

$$f''(x) = -\frac{2}{n}(\frac{A^2 + B}{(A + x)^3} + \frac{(n - A - C)^2 + D}{(n - A - x)^3}) \quad (6)$$

因为A、B、C、D均为大于等于 0 的常数，$x>0$，$n>n_l=A+x$，$(A+x)^3$和$(n-A-x)^3$是大于 0 的数，所以有$f''(x)<0$。根据定理 1 可知，$f(x)$在纯区间$[v_b, v_t]$上是上凸函数。因此式(3)在C_k的一个纯区间$[v_b, v_t]$内的极小值只可能出现在区间的边界点处。这样就只需要计算纯区间边界上的gini值，可以得到该纯区间的极小值，减小了计算量。

数值型属性一般都服从高斯分布[5]，在等宽划分的区间中存在大量的纯区间。因此，对于纯区间，只要计算各个纯区间的边界点处的gini值；对于非纯区间进行精确计算，就能得到整个属性值的最小gini值$gini_{low}$，找到最佳分裂点。

2.2 数值型属性的分裂方法

对数据集 S 中的某一个数值型属性分裂，分裂方法如下：

(1) 由于数值型属性的分类一般服从高斯分布，因此用等宽直方图方法将属性值分为 q 个区间，同时构造区间直方图列表。区间直方图列表的字段有区间的左边界、右边界的值和在该区间中各个类的记录数。由于区间直方图列表较小，它可以存放在主存中，提高计算效率。

(2) 对每一个区间计算gini值，并找出最小值$gini_{low}$：

1) 对于纯区间$[v_b, v_t]$，则计算区间右边界处的gini值$gini^D(S, a \le v_t)$；

2) 对于非纯区间，先建立区间属性表，再对区间进行排序，然后精确计算在该区间最小 gini 值。

(3) 用最小gini值$gini_{low}$对属性表进行分裂。

举例说明上述的分裂方法。设有一个数据集 S，如表 1 所示，对 Salary 属性进行分裂。若直方图的宽度为 30，可以将属性值域分为[1,30]、[31,60]和[61,90] 等 3 个区间，建立如表 2 所示区间直方图列表。

表 1 数据集

Age	Salary	Class	Tid
30	65	G	1
25	20	B	2
50	85	G	3
40	75	G	4
45	60	G	5
52	40	B	6
23	15	B	7

表 2 区间直方图列表

左边界	右边界	B	G
1	30	2	0
31	60	1	1
61	90	0	3

利用式(3)计算各个区间的 gini 值：

对于区间[1,30]，由于该区间是类“ B ”纯区间，计算该区间右边界处的gini值：$gini^D$(S,Salary≤30)＝0.228571；

对于区间[61,90]，由于该区间是类“ G ”纯区间，计算该区间右边界处的gini值：$gini^D$(S,Salary≤90)＝0.489796；

对于区间[31,60]，由于该区间是非纯区间，先对该区间的区间属性表进行排序并计算gini值：$gini^D$(S,Salary≤50)＝0；

最后得到Salary属性的最小gini值$gini_{low}$＝0，同时获得最佳分裂点为 50。

2.3 SPRINT 算法的改进

在改进的算法中，采用宽度优先的策略来构建决策树和使用 gini 指标来评估数值型属性。我们改进的主要部分就是在决策树的构建阶段对数值型属性的处理部分，采用了上述提出的纯区间归约的方法来处理数值型属性。改进算法的详细描述如下：

输入：训练集样本 T

输出：一棵决策树

步骤：

(1)T 满足停止扩展的条件，则返回

(2)对每个离散值属性

1)扫描属性列表，更新计数矩阵；

2)找到每个属性的最佳分裂子集。

(3)对每个连续值属性

1)等宽直方图方法将属性分为 q 个区间，建立分区直方图列表；

2)对每个纯区间计算边界处的 gini 值；

3)对每个非纯区间进行局部排序,计算该区间的极小 gini 值；

4)找出整个属性的最佳分裂点。

(4)比较各个属性的最佳分裂，选择一个最佳的将 T 分为 T1 和 T2。

(5)递归地对 T1 和 T2 生成决策树。

算法中，集合T、T_1、T_2分别代表树中的结点，其中T_1和T_2是T的两个分支结点。最后生成的决策树是一棵二叉树。

3 算法分析

由于改进的算法和 SPRINT 算法的不同之处主要是在构建决策树阶段对数值型属性的处理方法，算法的分析主要是比较两种算法在树的构建阶段的 I/O 需求和时间代价。我们的比较分为预处理阶段和决策树中每个结点的构建阶段。

假设数据集S有n条记录，它们分属于c个互不相关的类，划分为q个区间，每个区间的记录数为n_i；在改进的算法中非纯区间数为b。

(1)预处理阶段

SPRINT 算法中构建属性表需要一次读操作和一次写操作的时间代价为 $O(n)$；对每一个数值型属性预排序需要两次读和两次写操作的时间代价为 $O(nlogn)$。

改进的算法中构建属性表需要一次读操作和一次写操作，建立分区直方图列表需要一次读操作，下个预处理阶段的时间代价为 $O(n)$。

(2)结点构建阶段

在 SPRINT 算法中，计算一个数值型属性的最佳分裂点要对属性表进行一次读操作的时间代价为 $O(n)$；将属性表分裂需要一次读和一次写操作的时间代价为 $O(n)$。

在改进的算法中，估算每一个区间边界的 gini 值的时间代价为 $O(qc)$；决定每个非纯区间的记录，建立临时区间属性表需要一次读和一次写操作的时间代价为 $O(n)$；对每个非纯区间排序和计算每个非纯区间中的精确 gini 值的时间代价为 $O(\sum_{i=1}^{b} n_i \log n_i + n_i c)$；将属性表分裂需要一次读和一次写操作的时间代价为 $O(n)$。

在 SPRINT 算法中，对属性表的全部记录进行排序是整个处理过程的主要时间花销。在改进的算法中有效地避免了全局排序，只是对非纯区间进行局部排序；同时，纯区间进行归约，减小了 gini 值的计算量。

4 结论

改进的算法利用了 gini 指数函数在纯区间上是凸函数和数值型属性一般服从高斯分布的特点，对 SPRINT 算法在处理数值型属性部分进行了优化。算法分析表明，改进的算法是一个有效的处理方法。当然，划分的区间数对改进的算法的计算时间代价有一定的影响，如何有效地选取区间数，以便更好地减小非纯区间的数量及降低对非纯区间的排序代价，有待进一步的优化。

参考文献

1 Quinlan J R. C4.5: Programs for Machine Learning[M]. Morgan Kaufman, 1993.

2 Mehta M, Agrawal R, Rissancn J. SLIQ: A Fast Scalable Classifier for Data Mining[C]. Proc. of the 5th Int’l Conf. on Extending Database Technology, Avignon, France, 1996-03.

3 Shafer J, Agrawal R, Mehta M. SPRINT: A Scalable Parallel Classifier for Data Mining[C]. Proc. of the 22th Int’l Conf. on VLDB, Bombay, India, 1996-09.

4 Alsabti K, Ranka S, Singh V. CLOUDS: A Decision Tree Classier for Large Datasets[C]. Proc. of the 4th Int’l Conf. on Knowledge Discovery and Data Mining, 1998.

5 Han J, Kamber M. Data Mining: Concepts and Techniques[M]. Beijing: High Education Press, 2001: 279-301.

6 Wang H, Zaniolo C. CMP: A Fast Decision Tree Classier Using Multivariate Predictions[C]. Proc. of the 16th Int’l Conf. on Data Engineering, 2000.

7 Agrawal R, Ghosh S, Imielinski T, et al. An Interval Classifier for Database Mining Applications[C]. Proc. of the VLDB Conference. Vancouver, British Columbia, Canada, 1992-08.

8 Ruggieri S. Efficient C4.5[J]. IEEE Transactions on Knowledge and Data Engineering, 2002,14(2).

~~~~~~

(上接第 19 页)

(1)对象组相似度阈值 θ 的设定直接影响到聚类结果。θ 较小时，形成的对象组就比较多。在本实验中，取 θ=0.25。

(2)通过采用本章提出的聚类算法得出的对象聚类结果比较令人满意，且计算效率高。实验给出了一组测试数据，数据对象从 5 000 个开始，逐次增长到 100 000 个，对象的属性维为 100，系统运算时间如图 1 所示。

6 结论

本文介绍的聚类算法，首先定义了对象组相似度来进行对象集内部的相似程度计算。在保证不影响数据质量的前提下，通过采用对象组特征向量来描述对象组内部各个对象属性信息，有效地实现了对数据必要的压缩，减少了数据计算量。另外，该算法在整个执行过程中，只需要进行一次数据扫描，大大地提高了算法的效率。根据聚类结果，通过引入高维稀疏对象组的上确界和下确界表示，可以很方便地对新对象进行组分配。此方法不仅简单实用，得到的分类结果也是令人比较满意。

参考文献

1 史忠植. 知识发现[M]. 北京: 清华大学出版社, 2002.

2 Han J W. Data Mining[M]. Morgan Kaufmann Publishers, Higher Education Press, 2001.

3 许天周. 应用泛函分析[M]. 北京: 科学出版社, 2002.
~~~~~~