

# 基于 SPRINT 分类算法挖掘 保险业务数据中的风险规则

宾 宁

(广东工业大学 经济管理学院, 广东 广州 510520)

摘要: 提出利用 SPRINT 算法对保险业务数据进行风险分析. 针对医疗保险业务, 详细介绍了 SPRINT 算法的预处理、计算最佳分裂、执行分裂的具体设计实现过程, 并得出一些实用的风险规则.

关键词: SPRINT 算法; 分类算法; 数据挖掘; 保险业务

中图分类号: TP301.6      文献标识码: A      文章编号: 1007-7162(2007)02-0099-04

数据挖掘 (Data Mining) 就是应用一系列技术从大型数据库或数据仓库中提取人们感兴趣的信息和知识, 这些知识或信息是隐含的, 事先未知而潜在有用的, 提取的知识表示为概念、规则、规律、模式等形式<sup>[1-2]</sup>也可以说, 数据挖掘是一类深层次的数据分析. 近十几年来, 随着人们利用信息技术生产和收集数据的能力的大幅度提高, 数据挖掘被越来越多地应用于大中型企业、商业、银行、保险业和电信业各部门, 表现出极强的发展潜力<sup>[3]</sup>.

随着我国经济体制改革的深入发展和对外经济开放, 全社会的风险意识和保险意识不断增强, 保险作为风险管理的重要方法日益受到重视. 保险公司成功的一个关键因素是在设置具有竞争力的保费和覆盖风险之间选择一种平衡. 对于保险公司来说, 保险是一项风险业务, 其重要工作之一就是进行风险评估. 目前保险公司对风险的分析通常是粗略的或根据经验来判断的. 数据挖掘提供了进行保险投资组合数据库分析的环境, 数据挖掘的多种方法可用来进行风险分析<sup>[4]</sup>. 本文提出利用 SPRINT 分类算法, 在保险公司建立的客户信息数据库的基础上对客户的风险率进行判断, 帮助保险公司对其保险业务风险有一个准确的科学分析和预测.

## 1 SPRINT 分类算法简介

数据分类在数据挖掘中是一项非常重要的任务. 分类的目的是分析输入数据, 通过在训练集中的数据表现出来的特性, 为每一个类找到一种准确的描述或者模型, 用来对未来的测试数据进行分类<sup>[5]</sup>.

目前常用的分类算法有决策树分类算法、Bayes 分类算法、基于关联规则的分类算法以及基于数据库技术的分类算法等<sup>[6]</sup>. 其中, 决策树算法产生的规则易于理解, 决策树的每个分枝都对应一个分类规则, 而且其速度较快、准确率较高<sup>[7]</sup>, 因此, 常选择决策树算法来进行保险业务的风险分析.

SPRINT 算法是 John Shafer 和 Rakesh Agrawal 于 1996 年提出的针对大型数据库的一种高速可伸缩的数据挖掘分类算法<sup>[8]</sup>. 它能处理分类属性和连续值属性. SPRINT 算法使用预排序技术, 对非常大而不能放入内存的驻留磁盘的数据集进行预排序<sup>[9]</sup>. 为了减少需要驻留于内存的数据量, SPRINT 算法进一步改进了决策树算法实现时的数据结构, 将类别列合并到每个属性列表中. 这样, 在遍历每个属性列表寻找当前结点的最优分裂标准时, 不必参照其他信息. 而对节点的分裂表现在对属性列表的分裂, 即将每个属性列表分成两个, 分别存放属于各个节点的记录. 当表划分时, 表中记录的次序维持不变. 因此, 划分表不需要重新排序. SPRINT 的设计易于并行化, 这进一步增强了可伸缩性<sup>[10-11]</sup>.

## 2 应用 SPRINT 算法进行风险分析

本节将结合某医疗保险公司部分数据, 举例说明 SPRINT 算法在保险业务中的应用过程. 如表 1 所示, 该表记录了每个客户的 5 个属性信息. 选取 H i R i s k 作为目标属性, 建立一棵决策树, 用来帮助保险公司判断目标客户是否属于高风险客户.

表 1 某医疗保险公司部分数据集

| 序号<br>(R id) | 平均月访问量 /次数<br>(A vgM onth ly V isits) | 医疗记录<br>(M edicaH istory) | 平均每月医疗花费<br>(A vgM edical Expenses (M onth ly) ) | 收入<br>(I ncm e) | 是否高风险客户<br>(H iR isk) |
|--------------|---------------------------------------|---------------------------|--------------------------------------------------|-----------------|-----------------------|
| 1            | 0.8                                   | Bad                       | 53.00                                            | 39600           | Yes                   |
| 2            | 0.9                                   | Bad                       | 86.75                                            | 45500           | Yes                   |
| 3            | 1.0                                   | Excellent                 | 23.25                                            | 60100           | No                    |
| 4            | 0.8                                   | Good                      | 78.50                                            | 55000           | No                    |
| 5            | 0.7                                   | Good                      | 55.00                                            | 31300           | No                    |
| 6            | 0.6                                   | Bad                       | 65.50                                            | 37400           | Yes                   |
| 7            | 1.4                                   | Good                      | 28.25                                            | 44200           | No                    |
| 8            | 1.1                                   | Excellent                 | 33.75                                            | 28100           | Yes                   |
| 9            | 1.5                                   | Good                      | 14.25                                            | 25900           | No                    |
| 10           | 2.1                                   | Good                      | 39.75                                            | 38200           | Yes                   |
| 11           | 1.0                                   | Good                      | 27.00                                            | 35500           | No                    |
| 12           | 1.6                                   | Good                      | 31.25                                            | 41600           | Yes                   |
| 13           | 1.3                                   | Good                      | 29.00                                            | 57400           | Yes                   |
| 14           | 0.8                                   | Good                      | 46.00                                            | 47100           | No                    |
| 15           | 1.4                                   | Bad                       | 27.50                                            | 42300           | Yes                   |
| 16           | 1.0                                   | Good                      | 28.25                                            | 33000           | No                    |
| 17           | 1.2                                   | Good                      | 33.50                                            | 36800           | Yes                   |
| 18           | 1.1                                   | Excellent                 | 26.25                                            | 41300           | No                    |
| 19           | 1.1                                   | Good                      | 25.50                                            | 33500           | No                    |

数据分类工作包括以下几个步骤:

1)数据准备

系统输入是训练集 (Training Data). 训练集是样本向量  $(v_1, v_2, \dots, v_n; c)$ 组成的集合, 每个属性对应训练集的一列. 训练集进入之后, 分成一个一个的属性表 (Attribute List)  $\{(v_i, i) | i \leq \text{training data num} \& \& i \geq 0\}$ ,  $i$ 是属性  $v_i$ 的记录索引号. 对于连续值属性, SPRINT 算法首先要对每个属性列表进行内部预排序, 交换属性值  $v_i$ , 同时交换  $i$  生成有序的属性表序列 (以后可以看到, 这是 SPRINT 算法中对属性进行的唯一一次排序, 也是该算法的重要优点之一), 并将类别列合并到各个属性表序列中. 对于排序完后的属性表序列, SPRINT 算法采用直方图 (Histogram) 的数据结构进行存储, 它位于决策树的每个节点内, 分别用  $C_{\text{below}}$  和  $C_{\text{above}}$  来存放每个节点当前的类信息 (见图 1).

对于类别属性, 这种直方图还可以表示成计数矩阵 (CountMatrix) 的形式, 直观地记录每种属性值的类别分布情况 (见图 2).

完成属性列表的划分后, 数据的初始化工作就完成了.

2) 计算最佳分裂

完成数据预处理之后, 进入往复地求最佳分裂指标的阶段. 这一阶段, 经过一次对所有属性列表的遍历, 可以找出所有叶子节点的最佳分裂方案. 区别

于一般的决策树, SPRINT 采用二分查找树结构, 对每个节点都需要先计算最佳分裂方案, 然后执行分裂.

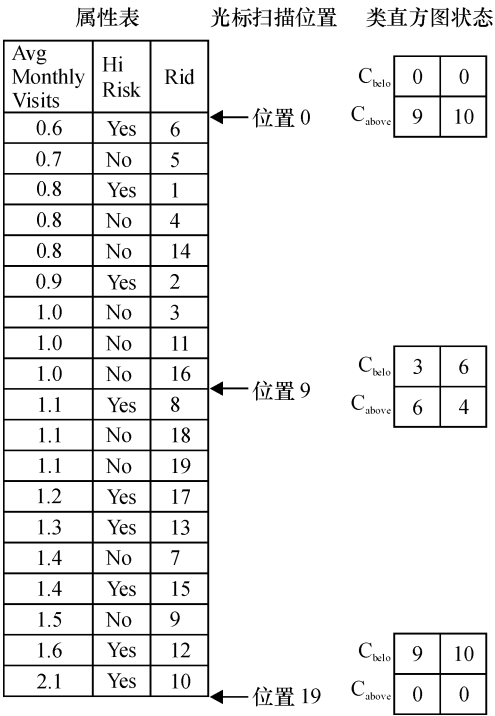


图 1 连续值属性 Avg Monthly Visits 的直方图表示

在选择当前节点的最优属性对数据集进行分割的这个环节中, SPRINT 算法采用基尼指数——Gini index 来度量最佳分裂点, 进行属性选择. 若集合  $S$  包含  $N$  个类别的记录, 那么其 Gini 指标就是

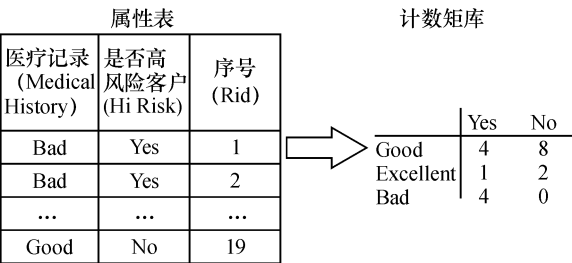


图 2 类别属性 Medical History 的直方图表示

$gini(S) = 1 - \sum p_j^2$ ,  $p_j$  是  $S$  中第  $j$  类出现的频率. 如果集合  $S$  分成两部分  $S_1$  和  $S_2$ , 那么这个分割的  $gini$  就是  $gini(S) = \frac{n_1}{n} gini(S_1) + \frac{n_2}{n} gini(S_2)$ .  $gini$  值越小, 表明信息增益量 (Information Gain) 越大, 节点分裂质量越好.

对于数值型连续属性 (continuous attributes) 分裂的形式  $A \leq v$  假设排序后的结果为  $v_1, v_2, \dots, v_n$ , 因为分裂只会发生在两个节点之间, 所以有  $n-1$  种可能性, 通常取中点  $(v_i + v_{i+1}) / 2$  作为分裂点. 从小到大依次取不同的 split point 取 Information Gain 指标最大 ( $gini$  最小) 的一个就是分裂点.

对于类别属性, 设  $S(A)$  为  $A$  的所有可能的值, 分裂测试将要取遍  $S$  的所有子集  $S'$ . 寻找当分裂成  $S'$  和  $S - S'$  两块时的  $gini$  指标, 取到  $gini$  最小的时候, 就是最佳分裂方法. 这是一个对集合  $S$  的所有子集进行遍历的过程, 共需要计算  $2^{|S|}$  次, 代价也是很大的, SPRNT 算法对此也有一定程度的优化.

要找到每个节点的最佳分裂属性, 则需要扫描每个属性列表, 分别计算各个属性的  $gini$  值, 然后选择  $gini$  值最低的作为该节点的最佳分裂属性. 计算最佳分裂的算法如下:

```
EvaluateSplits()  
{  
    for each attribute A do  
        traverse attribute list of A  
        for each value v in the attribute list do  
            if A is a numeric attribute then  
                compute splitting index for test(A <= v) for leaf node  
            if A is a categorical attribute then  
                for each leaf of the tree do  
                    find subset of A with best split  
        }  
}
```

以表 1 为例, 对于连续值属性 Avg Monthly Visits, 分别取每两个值的平均值计算其  $gini$  值. 如计算  $gini_{split}(0.65)$ :

$gini(S_1 \leq 0.65) = 1 - 1 = 0$

$gini(S_2 > 0.65) = 1 - [(8/18)^2 + (10/18)^2] = 40/81$

则  $gini_{split}(0.65) = 1/19 * 0 + 18/19 * 40/81 = 0.4678$

同理, 计算得

$gini_{split}(0.75) = 1/19 * 0 + 18/19 * 40/81 = 0.4985$

$gini_{split}(1.1) = 0.4311$ ,  $gini_{split}(1.35) = 0.4872$   
算出所有的  $gini$  值后, 选  $gini$  值最小的作为该属性分裂的最佳阈值, 对于 Avg Monthly Visits 属性, 则取分裂点 1.1.

对于离散型字段 History 属性, 它含有 good, excellent, bad 这 3 个属性值. 若取分裂点为 good 得

$gini(S_1 = \text{Good}) = 1 - [(1/3)^2 + (2/3)^2] = 4/9$   
 $gini(S_2 \neq \text{Good}) = 3/7 * gini(S_3 = \text{Excellent}) + 4/7 * gini(S_4 = \text{Bad})$   
 $= 3/7 * 4/9 + 4/7 * 0 = 4/21$

则  $gini_{split}(\text{Good}) = 12/19 * 4/9 + 7/19 * 4/21 = 0.3509$

同理, 可得  $gini_{split}(\text{Excellent}) = 0.3509$ ,  $gini_{split}(\text{Bad}) = 0.3509$

类似以上步骤, 可以得到各个属性的最佳分裂方案, 选择  $gini$  值最小的作为当前节点分裂属性, 则选择根节点为 Avg Medical Expenses 属性. 首次分类后的结果见图 3 属性 Avg Medical Expenses 具有最小  $gini$  值, 成为决策树根节点的测试属性.

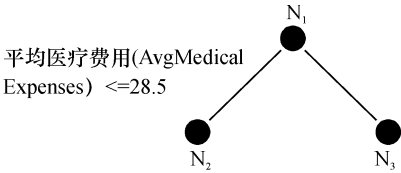


图 3 执行首次分裂后的决策树表示

3) 执行分裂

当最佳分裂参数已经存放在节点中后, 算法的下一步是执行分裂. 这一步的主要任务是将该分割点的属性列表分割, 并用该列表的 rids 生成记录所在节点的哈希表 (Hash Table), 用哈希表分割其他属性列表, 列表分割同时生成属性直方图.

3 决策树生成及分类规则分析

采用 Jbuilder9 开发工具, 结合 weka 开发平台实现 SPRNT 算法, 并对表 1 数据进行处理后得到完整决策树如图 4 所示.

沿着由根节点到叶节点的路径, 从以上决策树中可提取出以下分类规则:

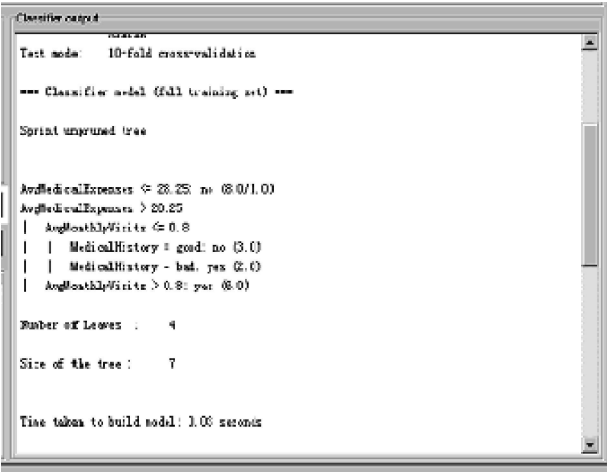


图 4 生成的保险数据集决策树

if Avg Medical Expenses < = 28.25 then Hi  
Risk= no

if Avg Medical Expenses > 28.25 and Avg  
Monthly Visits> 0.8 then HiRisk= yes

决策树给出的第一个选择条件为 Avg Medical  
Expenses 这与经验常识相符, 即: 医疗花费较低  
的人身体素质较好. 但具体到多少算医疗花费较低, 本  
例中决策树给出了一个量化指标 28.25 月医疗花  
费低于 28.25 的客户医疗风险率为 no 对于月医疗  
花费高于 28.25 的客户, 则还要考察第二个决策条  
件 Avg Monthly Visits 如果经常去医院看病, 则身  
体素质较差, 医疗风险相应较高, 本例中给出的分界  
线是 0.8

本例的数据集较小. 当数据集越大时, 得到的决  
策树准确度越高. 在系统中, 采用 SPR NT 分类算  
法将客户分为高风险客户和低风险客户两类. 因为不  
同的客户有着不同的投保需求, 保险公司应根据客  
户的需求模式对其进行分类, 提供相应的保险产品  
和服务, 更好地配置资源, 取得最大效益.

4 结论

本文介绍了 SPR NT 算法在挖掘保险业务数据  
中的具体应用过程. SPR NT 算法进一步改进了决策

树算法实现时的数据结构, 减少了需要驻留于内存  
的数据量, 这样, 在遍历每个属性列表寻找当前结  
点的最优分裂标准时不必参照其他信息. SPR NT 算  
法具有高并行性能, 可以处理庞大的训练集, 这使  
得它不仅适用于保险业, 还可用于银行、股票交易  
等投资领域以及商业等其他行业, 为管理人员控  
制投资风险, 取得更佳业绩提供有力的决策支持.

参考文献:

[ 1 ] Jiauei Han, Micheline Kamber 数据挖掘概念与技术  
[ M ]. 范明, 孟小峰译. 北京: 机械工业出版社, 2001.  
[ 2 ] Jiawei Han Micheline Kamber Data Mining Concepts and  
Techniques[ M ]. Simon Fraser University, 2000: 187-198  
[ 3 ] Agrawal R, Imielinski T, Svm i A. Database mining: A per-  
formance perspective[ J]. IEEE Trans On Knowledge and  
Data Engineering 1993, 5( 6): 914-925  
[ 4 ] 李波. 基于 SLIQ 分类算法的数据挖掘技术及其在企业  
CRM 中的应用 [ J ]. 计算机工程与应用, 2002, 21: 29-  
31  
[ 5 ] 刘红岩, 陈剑. 数据挖掘中的数据分类算法综述 [ J ]. 清  
华大学学报: 自然科学报, 2002, 42( 6): 727-730  
[ 6 ] J R Quinlan Induction of decision trees[ J ]. Machine  
Learning 1986 1( 1): 81-106  
[ 7 ] Bowyer K. A Parallel decision tree builder for mining very  
large visualization datasets[ C ] // Proceedings of the IEEE  
International Conference on Systems Man and Cybernetic,  
2000( 3): 1888-1893  
[ 8 ] John Shafer Rakesh Agrawal SPR NT: A scalable parallel  
classifier for data mining[ C ] // Proceedings of the 22nd  
VLDB Conference Mumbai( Bombay). India 1996: 544-  
555  
[ 9 ] Rakesh Agrawal Sakti Ghosh Tomasz Imielinski et al An  
interval classifier for database mining application[ C ] // In  
Proc of the VLDB Conference Vancouver British Columbia  
Canada 1992: 560-573  
[ 10 ] 郭景峰, 米浦波. 决策树算法的并行性研究 [ J ]. 计算机  
工程, 2002, 28( 8): 77-78.  
[ 11 ] James M. Classification Algorithms[ M ]. New York: John  
Wiley & Sons, 1985.

Risk Analysis of Insurance Business Based on SPR NT Classification Algorithm

BN Ning

( Faculty of Economics and Management, Guangdong University of Technology, Guangzhou 510520, China)

**Abstract** Risk evaluation is very important to insurance company. This paper presents an insurance business risk analysis method based on SPR NT classification algorithm. Aiming at the medical insurance instance, it introduces the implementation of SPR NT in details, and then mines some rules which could be utilized in the field of insurance.

**Key words** SPR NT; classification algorithm; data mining; insurance business