# 基于规则的市场分析方法*

马少平　郑　彤　陆玉昌

（清华大学计算机科学与技术系　智能技术与系统国家重点实验室　北京　100084）

**摘要**　本文叙述了基于规则的市场分析方法，结合随机性市场分析的特点，讨论了规则应具有的表达能力，引用S—表达式的概念，建立了归一化存储结构，提出了变量环境的概念，实现了合一过程，建造了一个基于规则的市场分析报告生成系统.

**关键词**　规则基系统，市场分析，报告生成.

市场分析一般包括2部分内容，① 对市场上有关产品竞争态势及生产厂家情况的分析，如厂家产品的市场占有率，销售价格，与厂家产品构成竞争的品牌销售、价格情况及其市场走势等，这一部分内容属于对市场的一般性经济情况的分析，称为常规性分析. ② 对市场上突发事件的分析，如某产品价格突然大幅度上升或下降，近一个时期连续上升或下降，某产品在某些省、市市场占有率出现大幅度变化等，当出现这些突发事件时，生产厂家希望知道相关产品的价格、市场占有率的变化情况、造成这些现象的原因及事件发生地的各种经济指标情况等. 由于突发事件不是经常发生的，具有一定的随机性，故可称为随机性分析.

本文在总结了经济专家进行随机性市场分析经验的基础上，建造了一个基于规则的随机性市场分析系统，该系统对数据库中的市场数据进行综合分析，找出市场上的突发性事件，对其发生的背景和相关性内容进行分析，按照突发性事件发生的强度，分出不同的等级，最后以文字、简表、卡片和统计图形等多种形式给出随机性市场分析报告，供决策者进行市场决策.

本文第1节介绍规则应具有的表达能力，第2节介绍系统的构成框图，第3节介绍系统的实现方法及原理，第4节总结.

## 1　对规则表达能力的要求

系统中知识采用产生式规则的表达方式，由于市场分析的复杂性，涉及逻辑性知识、过程性知识和计算性知识等多种知识形式，对规则的表达能力和处理能力具有较高的要求：

(1) 为获取数据，须具有查询数据库的能力，也就是说，当知识涉及到市场数据时，可以很方便地在规则中表达出来.

---

* 作者马少平，1961年生，副教授，主要研究领域为知识工程，汉字识别. 郑彤，1971年生，硕士生，主要研究领域为知识工程. 陆玉昌，1937年生，教授，主要研究领域为知识工程，机器学习.

本文通讯联系人：马少平，北京 100084，清华大学计算机科学与技术系

本文1996－01－30收到修改稿

(2) 为适应市场分析中的复杂计算，规则须具备与 C 语言的接口能力，用户可以很方便地将 C 函数与规则结合起来.

(3) 分析中经常出现递归性很强的知识，因此规则必须支持递归定义.

(4) 分析结果要以文字、简表、卡片和统计图形等多种形式综合给出，为做到报告的"拟人化"，须具有对结果的归纳能力、自然语言生成能力和相应的简表、卡片、统计图形的生成能力.

## 2 系统构成框图

系统构成框图如图 1 所示. 其中原始数据存放于 Sybase 数据库中，数据预处理部分根据预处理描述表中描述的方法对原始数据进行规范化处理后，形成规范化数据存于 Sybase 数据库中(请参见本辑郑彤一文). 推理机在规则的引导下对规范化后的数据进行处理，从中找出市场上的突发事件，形成报告生成用大纲，同时记录相关的推理信息及突发事件的数据来源，产生解释用信息，报告文本生成和简表、卡片、统计图形生成部分根据推理机提供的报告大纲，对大纲进行归纳和整理，最终生成出配有简表、卡片、统计图形的文字报告，而解释文本生成部分则根据解释用信息产生出解释文本，供以后查询使用.

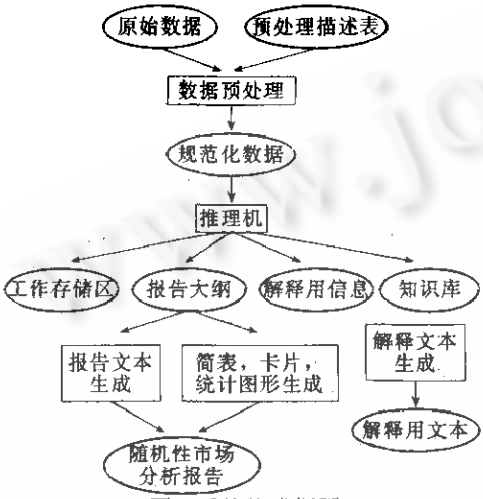


图1 系统构成框图

## 3 原理与实现

(1) 归一化存储结构

为适应知识表达中数据类型的多样性和不可预见性，我们借用了 LISP 语言中 $S$—表达式的概念.

**定义** 1. 浮点数、整数和符号统称为原子，其中符号包括英文符号和汉字符号. 每个符号原子可以具有一系列的属性和值.

**定义** 2. $S$—表达式定义如下：

① 原子是 $S$—表达式.

② 如果 $A$ 与 $B$ 均是 $S$—表达式，则点对$(A \cdot B)$也是 $S$—表达式. 其中 $A$ 称为该点对的头，$B$ 称为它的尾.

**定义** 3. 表是一种特殊的 $S$—表达式. 对于如下形式的点对：

$$(a_1 \cdot (a_2 \cdot \cdots \cdot (a_n \cdot NIL)))$$

可以简写为$(a_1\ a_2 \cdots\ a_n)$，并称为表，其中 $NIL$ 表示"空". 这里 $a_1$ 是该表的头，$(a_2 \cdots\ a_n)$是它的尾.

我们定义了一个如下形式的归一化存储结构，来表达 $S$—表达式：

```
typedef struct node
{   union
```

```
{
  struct node *node_ptr;
  char *string;
  int i;
  float f;
} head;
union
{
  struct node *t;
  P_FUNC p;
  FILE *fptr;
} tail;
char type;
} NODE;

typedef struct node *(*P_FUNC)(struct node *);
```

其中 *type* 表示 $S$－表达式的类型,取值及其含义如下:

'*i*':整型,*node.head.i* 为其取值.

'*r*':实型,*node.head.r* 为其取值.

'*c*':常量符号,*node.head.string* 为其符号串,*node.tail.t* 为其属性表指针.

'*v*':变量符号,*node.head.string* 为其符号串,*node.tail.t* 为其属性表指针.

'*p*':函数指针,*node.tail.p* 为其指针值,*node.head.string* 为函数名.

'*l*':表,*node.head.node_ptr* 为表头指针,*node.tail.t* 为表尾指针.

'*f*':文件指针,*node.tail.fptr* 为其指针值,*node.head.string* 为文件名.

图 2 给出的是用上述存储结构存储表$(a,b,c)$时的示意图.

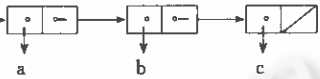


图2 表(a b c)的存储示意图

(2) 变量与环境

为处理方便,并把汉字作为变量处理,我们约定以"?"开头的符号为变量.规则中出现的变量,全部被视为局部变量.这样,不同规则中出现的变量,即使是名字相同,也应被视为不同的变量.即便是同一规则中的同一变量,当该规则处于不同的调用环境时,如在递归调用的不同层次上,也应被视为不同的变量.为此,我们引入概念—"环境"来解决这一问题.

**定义** 4. 环境是如下形式的表:

```
((环境号)((变量名 1)(值 1)(值 1 所在环境))
         ((变量名 2)(值 2)(值 2 所在环境))
         ...
         ((变量名 n)(值 n)(值 n 所在环境)))
```

环境是在推理过程中动态生成的,每进行一次规则调用,就要形成一个新的环境.变量所处的环境由环境号标识,不同环境下的变量均是互不相同的变量,只有同一环境下的同名变量才是相同的变量.

**定义** 5. 如果一个变量在它所在的环境中没有值,则称其为自由变量,或者虽然有值,但其值是一个自由变量,则该变量也是自由的,否则称该变量为一个约束变量,其值为该变量的约束值.

(3) 合一算法

规则的匹配通过合一来完成,这里采用最基本的算法进行合一,其过程如下:

```
int unify (NODE *obj1, NODE *obj1_env, NODE *obj2, NODE *obj2_env)
/* 欲对 obj1 和 obj2 进行合一,其中 obj1_env 和 obj2_env 分别为
   obj1 和 obj2 所在的环境 */
{
  NODE *value, *value_env;
  if (eql(obj1, obj2) && eql(obj1_env, obj2_env)) return 1;
  /* 如果 obj1 与 obj2 相等,且它们所在的环境也相等,则 obj1 与 obj2 是可合一的 */
  else if (varp(obj1) && bound_info(obj1, obj1_env))
    {
      value=bound_value(obj1, obj1_env, &value_env);
      return unify(value, value_env, obj2, obj2_env);
      /* 如果 obj1 是一个约束变量,value 是其约束值, value_env 是其约束值所在环境,则只有当 value 与 obj2 可合
      一时, obj1 与 obj2 才是可合一的 */
    }
  else if (varp(obj2) && bound_info(obj2, obj2_env))
    {
      value=bound_value(obj2, obj2_env, &value_env);
      return unify(obj1, obj1_env, value, value_env);
      /* 如果 obj2 是一个约束变量,value 是其约束值, value_env 是其约束值所在环境,则只有当 obj1 与 value 可合
      一时, obj1 与 obj2 才是可合一的 */
    }
  else if (varp(obj1))
    {
      bind(obj1, boj1_env, obj2, obj2_env);
      return 1;
      /* 如果 obj1 是一个自由变量,则 obj1 与 obj2 可合一,且合一后 obj1 的约束值为 obj2 */
    }
  else if (varp(obj2))
    {
      bind(obj2, obj2_env, obj1, obj1_env);
      return 1;
      /* 如果 obj2 是一个自由变量,则 obj1 与 obj2 可合一,且合一后 obj2 的约束值为 obj1 */
    }
  else if (atom(obj1) || atom(obj2)) return eql(obj1, obj2);
  /* 当 obj1 或 obj2 为常量原子时,只有它们是相同的原子时,才是可合一的 */
  else if (unify(car(obj1), obj1_env, car(obj2), obj2_env))
    {
      return unify(cdr(obj1), obj1_env, cdr(obj2), obj2_env);
      /* 当 obj1 和 obj2 均为表时,只有当它们的头和尾分别可合一时,obj1 与 obj2 才是可合一的 */
    }
}
```

以上过程是递归定义的,当 2 个 $S$-表达式 *obj*1 和 *obj*2 可合一时,返回值 1,否则返回值 0,在判断是否可合一的过程中,如果 *obj*1 与 *obj*2 可合一,则将使二者合一所进行的置换约束于它们所处的环境之中.

(4) 推理机制与知识表示

根据随机性市场分析的特点,推理机制采用确定性的正、逆向混合推理方法.市场会出现哪些类型的突发事件,由用户在建造知识库时用规则事先定义好,在推理过程中,系统在这类规则的引导下,通过对规则集的匹配和数据查找,验证各类突发事件是否发生,这是一个逆向过程.当发生了某个突发事件时,则对其产生的背景和相关条件进行跟踪,找出该突发事件发生的原因和背景,这是一个正向的过程.

知识采用产生式规则表示,其格式如下:

如果 {〈条件〉}$^{+}$

则　〈结论〉

其中〈条件〉和〈结论〉均为表.

(5) 解释的生成

系统提供了2种类型的解释功能.一种是服务于系统维护人员的,它可以提供得出某项结论的推理路径、所用规则及内容以及推理依据等内容.系统维护人员可以据此对知识库的正确性进行维护.这种解释通常是专家系统意义下的解释,只要在推理过程中记录下推理路线和所用到的数据就可以了.另一类解释是服务于生产决策人员的,他们关心的是某项突发事件产生的原因和背景.由于篇幅的限制,分析报告中不能包括所有突发事件的全部信息,当决策人员想知道某个突发事件的更详细情况时,可以通过解释功能得到.由于该类信息量较大,在正常推理情况下并不保留这些信息,只有当用户要求对某项内容提供更详细情况时,系统才重新启动推理机,将有关更详细的情况报告给用户.

(6) 外部接口

系统分别建立了与Sybase数据库和C语言的2种外部接口,供在规则中使用.与Sybase的接口有2种,其一为检索数据库,调用格式如下:

(检索〈结果〉〈表名〉〈返回域〉|({〈返回域〉}$^{+}$)〈检索条件〉)

其功能为,按〈检索条件〉对名为〈表名〉的数据表进行检索,满足检索条件的记录,其〈返回域〉的值约束给〈结果〉,其中〈结果〉为一变量.

与Sybase的另一个接口为排序,其调用格式为:

(排序〈结果〉〈名次〉〈表名〉〈排序域〉〈返回域〉|({〈返回域〉}$^{+}$)〈排序条件〉)

其功能为,对名为〈表名〉的数据表中满足〈排序条件〉的记录按照〈排序域〉对其进行排序,将〈返回域〉的值及其排序次序约束给〈结果〉和〈名次〉,其中〈结果〉和〈名次〉均为变量.

与C的接口格式如下:

(*C_func* 〈结果〉(〈C函数名〉{〈参量〉}$^{*}$))

其功能为用参量表{〈参量〉}$^{*}$对名为〈C函数名〉的函数进行调用,该函数的返回值约束给〈结果〉,其中〈结果〉为变量,这里的C函数须具有以下格式的定义头:

*NODE* * 〈C函数名〉(*NODE* * *parm_list*)

(7) 报告生成

报告生成包括文字报告和简表、卡片、统计图形等项内容,该部分请参见本辑郑彤一文.

## 4　结　语

基于以上原理,我们为某企业建立了一个随机性市场分析报告生成系统,系统用C语

言编写，运行于 SGI 工作站上，经初步测试，系统的分析结果已达到了专家的分析水平，能够满足生产实际的需要.

**致谢** 本文是在石纯一教授指导下完成的，在系统的建造过程中，得益于石教授的许多具体指导和建议，在此特表谢意.

## 参考文献

1 林尧瑞，马少平. 人工智能导论. 北京：清华大学出版社，1989.
2 林尧瑞，张钹，石纯一. 专家系统原理与实践. 北京：清华大学出版社，1988.
3 林尧瑞，陆玉昌，马少平. IBM－PC 机上的人工智能语言－GCLISP 和 Micro_PROLOG. 北京：清华大学出版社，1990.
4 刘建国. 计量经济分析软件包 Micro TSP 使用指南. 北京：清华大学出版社，1991.
5 黎诣远. 微观经济分析. 北京：清华大学出版社，1987.

# A METHOD FOR MARKET ANALYSIS BASED ON RULES

Ma Shaoping Zheng Tong Lu Yuchang

(*Department of Computer Science and Technology Tsinghua University Beijing* 100084)

**Abstract** In this paper, a method for the market analysis based on the rules is presented and the expressing ability of the rules is discussed in conjunction with the features of random market analysis. The concept of S—expression is used to establish a unified store structure. The concept of variable environment is brought forward and a unified algorithm is realized. Finally, a rule—based report generation system for market analysis is build up.

**Key words** Rule—based system, market analysis, report generation.