一种海量规则模式匹配方法

张桂刚

（清华大学信息技术研究院 北京 100084）（湖北经济学院信息管理学院 武汉 430205）

摘 要 基于各种海量规则信息处理的需求，提出了一种海量规则模式匹配方法。设计了海量规则模式匹配方法的基本算法步骤，研究了各种规则节点的匹配处理方法。最后总结了海量规则模式匹配方法的特点。海量规则模式匹配算法部分拓展了现有规则匹配处理模式，提出了新的匹配处理方法。对比结果表明，该方法具有较好的效果。

关键词 规则，模式，模式匹配，海量规则网，规则节点

中图法分类号 TP301.6 **文献标识码** A

A Kind of Pattern Matching Method of Mass Rules

ZHANG Gui-gang

(Research Institute of Information Technology, Tsinghua University, Beijing 100084, China)

(School of Information and Management, Hubei University of Economics, Wuhan 430205, China)

Abstract Based on requirement of a large sale of rules information processing, a pattern matching methods of mass rules was proposed. Mass rule pattern matching based algorithms steps and all kind of rule nodes' matching processing methods were researched in order to improve processing efficiency. Finally, mass rules pattern matching methods' characteristics were summarized. Mass rules pattern matching methods extended the existing mass rules pattern matching processing pattern and proposed new processing method. Comparative results show that the method has good effect.

Keywords Rules, Pattern, Pattern matching, Mass rule network, Rule node

规则处理引擎作为产生式系统的一部分，当进行事实判断时，包含3个阶段：匹配、选择和执行。传统的规则模式匹配处理算法中，以RETE算法、TREAT算法以及LEAPS算法3种最为典型。在这些基本的规则处理算法基础上已经产生了Drools、ILOG、RUBIC等各种规则系统。研究人员在上述几种最典型的规则处理算法基础上，从不同的角度研究出各种不同的规则处理算法，主要有FP-tree[1]、Lana-match[2]以及其他数据驱动[3]的规则处理算法。不过，若不考虑RETE算法需通过冗余[4]大量规则节点来达到提高效率的目的，其仍然是传统规则处理算法中最好使用的一种算法。

后来，由于规则数量的不断扩大，为了能够处理较大规则的规则数量，很多研究者在传统规则处理算法，尤其是RETE算法等基础上做了进一步探讨，它们有些能够检测比传统RETE更为复杂的触发事件[5]。设计了一些并行产生式系统[6]的规则处理算法，处理流量较大的、面向集合的简单规则处理原型。

传统的规则处理算法主要存在如下问题。

(1)传统规则模式匹配算法中的规则节点语义性比较简单。不管是RETE算法、TREAT算法、LEAPS算法，还是其他传统的规则处理算法，它们都有一个共同的缺陷，就是模式匹配算法中的规则节点范围单一，只有模式节点与连接节点两种，当遇到语义性比较复杂的需求时，没法灵活处理。

(2)传统规则模式匹配算法语义共享性不够。

正是因为传统规则处理模式匹配算法（RETE算法、TREAT算法、LEAPS算法或者其他传统的规则处理模式匹配算法）没能体现复杂语义性的复杂规则节点，导致这些算法的共享节点范围太少。这些传统的规则处理模式匹配算法在设计时本身考虑不够复杂，只有模式节点与连接节点两种，因此它们只有模式节点（也叫单输入节点）及其连接节点（双输入节点）可以实现共享。然而，随着规则数量的无限扩大（上亿级别规则），以及语义复杂性需求越来越多，现有的规则处理算法很难适应这些需求。

本文针对上述不足，提出了一种新的海量规则模式匹配方法，目的在于解决目前规则引擎不能满足各种粒度节点设置规则以及不能针对海量规则形成规则网并进行优化处理的缺陷。新的海量规则模式匹配方法将极大扩展语义性，在规则处理理论及在物联网、电子商务、电子政务及各类预警系统规则处理应用中具有重大的理论及其应用意义。

1 模式网与海量规则网

假设有模式网络Relation_1（见表1）与一个已经存在的海量规则网（见图1）。

假设海量规则网[8]的部分局部由规则关系节点、规则选择节点、规则动作节点以及规则联合节点等规则节点组成。

到稿日期：2011-02-05 返修日期：2011-06-24 本文受2011年度湖北省教育厅人文社会科学研究项目（2011jytq055），国家973计划项目（2011CB302302），国家自然科学基金（60873007）资助。

张桂刚（1978—），男，博士后，讲师，CCF会员，主要研究方向为云存储、规则计算、语义计算、云计算，E-mail：zhangguigang@163.com。

表 1　模式网络 Relation_1

	属性 1	属性 2	属性 3	属性 4	属性 5
1	●	●	●	●	●
2	●	●	●	●	●
3	●	●	●	●	●
4	●	●	●	●	●
5	●	●	●	●	●

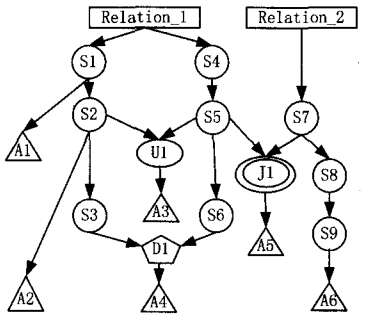

图 1　海量规则网局部

为了更好地分析海量规则的处理，首先介绍它的第一个算法，即海量规则模式匹配算法。

2　海量规则模式匹配算法

步骤 1　匹配海量规则网的规则关系节点与模式网中的模式。若匹配成功，则继续匹配相应的关系节点表下的所有选择节点与模式网中的所有事实。若匹配不成功，则将该关系表及其表下所有节点的出度流量均置为 0。同时计算匹配成功的海量规则网中关系节点的出度流量。

图 2 中，在匹配海量规则网的关系表节点与模式网中的模式的过程中，海量规则网的关系节点有两个，分别为 Relation_1 与 Relation_2。而在模式网络中，只有 Relation_1 这样一个模式匹配成功。计算出 Relation_1 关系节点的出度流量，它总共有两个出度，均将其置为 *R1_out*。而 Relation_2 模式没有匹配成功，故将海量语义关系网中的关系节点 Relation_2 及它之下的所有节点的出度均置为 0。

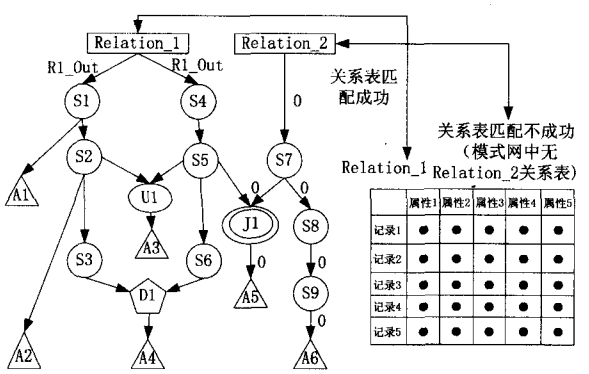

图 2　海量规则模式匹配算法步骤 1

步骤 2　若步骤 1 匹配成功，则选择该关系表下的所有选择节点与模式网络中的所有事实进行匹配。若再次匹配成功，则计算各个匹配成功的规则节点的出度流量，并将所有匹配不成功的节点的出度流量置为 0。

在图 3 中，首先从海量规则网中选择匹配成功的关系节点下的所有选择节点（可以将选择节点存储为一个数组），然后逐个地与关系表中的所有事实进行比较。通过反复循环比较运算，得出 S1，S2，S4，S6 分别与关系表 Relation_1 中的相应区域的事实符合要求。于是立即计算这些能够与事实相符合的节点的出度流量。S1 的两个出度流量相等，均为 *S1_out*；S2 的 3 个出度流量相等，均为 *S2_out*；S4 有一个出度，其流量为 *S4_out*；S6 也有一个出度，其流量为 *S6_out*。而匹配过程中，在关系表里没有找到相应匹配事实的选择节点有 S3，S5 两个，将其出度流量置为 0。

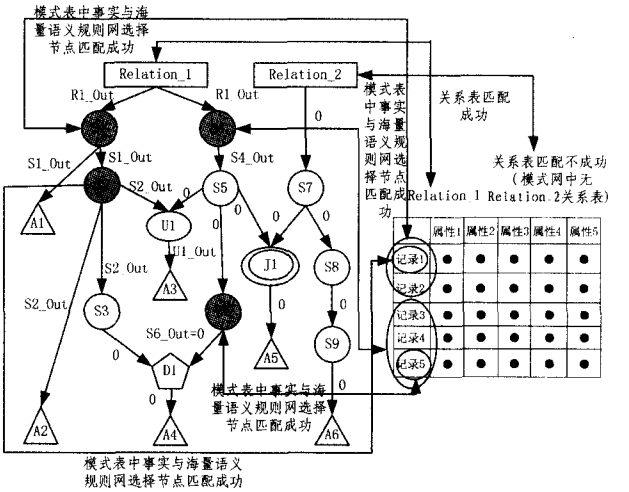

图 3　海量规则模式匹配算法步骤 2

这里的出度流量就是通过选择筛选后关系事实表里的满足条件的数据记录的实际条数，即

$$\begin{cases} S1\_out=2 \\ S2\_out=1 \\ S3\_out=0 \\ S4\_out=3 \\ S5\_out=0 \\ S6\_out=1 \end{cases}$$

匹配之后，有如下结果。

• 满足节点 S1 筛选条件的记录条数为 2 条，是图中的记录 1 与记录 2。

• 满足节点 S2 筛选条件的记录条数为 1 条，是图中的记录 1。

• 满足节点 S3 筛选条件的记录条数为 0 条。

• 满足节点 S4 筛选条件的记录条数为 3 条，分别是图中的记录 3、记录 4 与记录 5。

• 满足节点 S5 筛选条件的记录条数为 0 条。

• 满足节点 S6 筛选条件的记录条数为 1 条，是图中的记录 5。

步骤 3　计算所有其他节点（除关系节点与选择节点之外的所有节点）的出度流量（见图 4）。

在海量规则网的关系节点 Relation_1 下的所有节点，选择节点已经全部处理完毕，只剩下一个联合（Union）节点 U1 与一个否定（Denial）节点 D1。分别调用联合节点的出度流量计算算法（本文不讨论该算法）与否定节点出度流量计算方法（本文不讨论该算法），可以得出

$$\begin{cases} U1\_out=S2\_out=1 \\ D1\_out=0 \end{cases}$$

步骤 4　所有动作节点的入度流量大于 0 的所有规则（规则中所有条件均已经满足）均放到规则执行集中，进行规则触发。

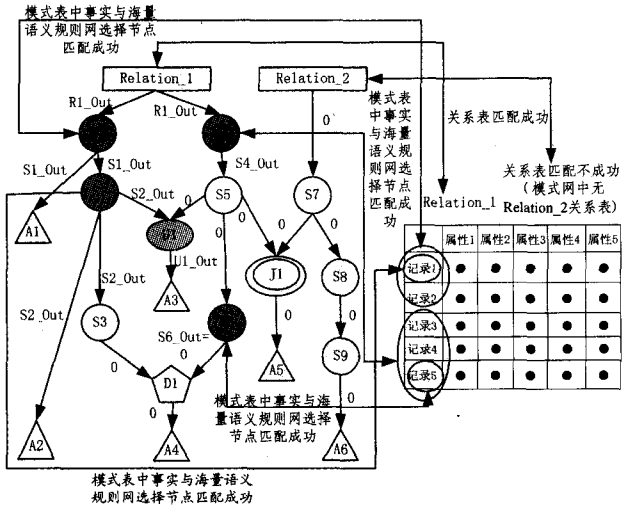

图 4　海量规则模式匹配算法步骤 3

经过上述步骤后，动作节点 A1，A2，A3 的入度流量均大于 0，也就是说关系表里的事实有满足规则节点条件的记录。于是，将 A1，A2，A3 所对应的规则 1、规则 2、规则 3 分别放入规则执行集中，触发它们，并将触发之后执行的动作结果发送给相应的用户。

步骤 5　动作节点的入度等于 0 的所有规则（规则中部分条件不满足或者全部条件均不满足）仍然处于非激活状况，等待发生事实条件的改变，即所有的规则仍然处于不被激活状况。

同理，经过上述步骤后，动作节点 A4，A5，A6 的入度流量均为 0，也就是说关系表里的事实没有满足规则节点条件的记录。于是，将 A4，A5，A6 所对应的规则 4、规则 5、规则 6 仍然置于非激活状况，继续等待关系表里的事实数据的发生（见图 5）。

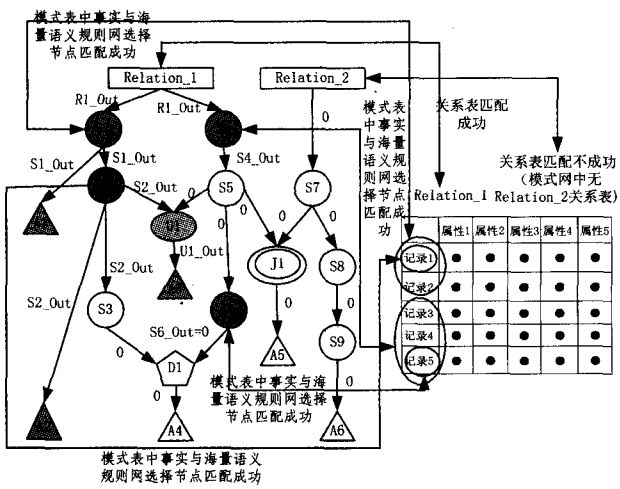

图 5　海量规则模式匹配算法步骤 4

步骤 6　算法完毕。

本算法中的海量规则网包含 n 个节点，即节点的数量为 n。假设用户设定的所有规则中条件部分的个数为 m，算法时间复杂度主要消耗在对规则节点与条件个数的比较上，比较次数为 $m*n$，其总的时间算法的复杂度为 $O(mn)$。

3　海量规则模式匹配算法的特点

(1)海量规则模式匹配算法与传统的规则模式匹配算法的相同点在于：

• 它们都有一个模式网络。所谓模式网络，就是关系表的所有事实构成的一个网络。

• 只要模式网络中的事实与规则的条件中所要求的事实相符合，就会触发相应的规则，并做出预先设计好的执行动作。

(2)海量规则模式匹配算法与传统规则模式匹配算法的最大区别就是：

• 传统的规则模式匹配算法基本上都建立在 RETE 算法的基础上，本质上没有很大的改变。海量规则模式匹配算法除了保留与传统规则网络相同的模式网外，整个处理算法发生了很大改变，它不需要事实表与一条条规则的条件进行比较，只需要与海量规则建立的海量规则网中的规则节点进行比较。

• 传统规则模式匹配算法需要模式网与连接网组成。而海量规则模式匹配算法中涉及到模式网与海量规则网。

• 传统规则模式匹配算法只能处理、选择双输入节点（包括连接节点与否定节点两种节点）。而海量规则模式匹配算法可以处理多种节点，它不仅能够处理传统规则模式匹配算法的模式节点（关系节点）和连接节点（Join），也能够处理更为复杂的规则节点，如规则联合节点（Union）、规则交集节点（Intersection）、规则否定计算节点（Denial）、规则笛卡尔积节点（Cartesian Product）等。

• 传统规则处理算法将所有规则中条件部分与现有事实进行匹配，若事实成立，则规则成立。而海量规则模式匹配算法将现有事实与海量规则网中各种事实节点（不需要与其他节点进行比较）进行匹配，若节点匹配成功，则触发相应的规则（规则集）。

4　结论分析

本文选择了传统的经典算法 RETE 进行类比。由于传统的经典算法 RETE 主要针对选择节点与双输入节点的连接节点（Join 运算）进行处理。通过类比与实验得出表 2 所列结论。

表 2　新算法与 RETE 处理节点的能力比较

	Select	Join	Union 等其他节点
RETE	能处理	能处理	不能处理
新算法	能处理	能处理	能处理
结论对比	效果一样	效果一样	不能对比

针对 RETE 与新算法的 Select 操作与 Join 操作，由于两种算法都只是针对数据库进行同样 SQL 操作，因此它们的效率应该相等。对于 Union 等其他节点，RETE 没法处理，新算法无法与之类比。

结束语　本文首先介绍了传统的规则模式匹配算法 RETE，TREAT，LEAPS 以及其他各种目前大多数规则引擎所使用的规则处理算法。尤其以卡内基梅隆大学 Forgy[7] 博士在他的博士论文中所阐述的 RETE 算法最为经典。传统规则模式匹配算法所处理的规则语义性不强，只能处理选择节点以及连接节点两种主要节点，很难满足用户设定规则的语义性需求。为了弥补传统规则处理算法的各种缺陷，本文提

（下转第 177 页）

100 次聚类算法后各项评价指标的平均值、最小值、最大值和标准差。

表 2 在 Chess 下的性能比较

	Before FS			After FS		
	AC	PE	RE	AC	PE	RE
Mean	0.55018	0.55466	0.68572	0.56277	0.77025	0.9487
Min	0.52222	0.26111	0.5195	0.52222	0.76103	0.82504
Max	0.70651	0.70783	1	0.66051	0.7923	0.9997
SD	0.03645	0.05084	0.09167	0.06281	0.01417	0.07945

表 3 在 Lung-Cancer 下的性能比较

	Before FS			After FS		
	AC	PE	RE	AC	PE	RE
Mean	0.72625	0.72956	0.69611	0.74219	0.75265	0.7493
Min	0.71875	0.61677	0.5604	0.71875	0.63462	0.5386
Max	0.8125	0.89655	0.922	0.8437	0.89655	0.9782
SD	0.01546	0.02862	0.07113	0.0408	0.04956	0.08576

表 4 在 Soybean 下的性能比较

	Before FS			After FS		
	AC	PE	RE	AC	PE	RE
Mean	0.86702	0.90311	0.91462	0.84617	0.89684	0.9305
Min	0.68085	0.7333	0.70735	0.5744	0.56136	0.84118
Max	0.97872	0.97727	0.985	0.9787	0.9773	0.9853
SD	0.101	0.0656	0.07085	0.1141	0.08795	0.05102

通过分析表 2－表 4 可以看出，数据集 Chess 和 Lung-Cancer 在通过特征选择后的特征子集上的聚类精度得到了显著提高；数据集 Soybean 的召回率 *RE* 得到了提高。综上，实验结果表明，算法 1 可以选择到有效的具有聚类性质的特征子集，而且有效提高了数据集的聚类精度。

结束语 本文基于分析数据集中特征的重要度与特征在数据集上的分类能力之间的关系，提出了一种基于邻域距离的特征选择方法，在得到的特征子集上可对数据集进行重新聚类。实验表明，该方法不仅可以选择出具有聚类性质的有效特征子集，而且重新聚类后，数据集的聚类精度得到了显著提高，并降低了针对高维复杂数据集聚类的计算耗时。本文为高维复杂数据集的聚类提供了新的方法和视角。

参 考 文 献

[1] Han Jia-wei, Kamber M. Data Mining Concepts and Techniques [M]. San Francisco: Morgan Kaufmann, 2001

[2] Brendan J F, Delbert D. Clustering by passing messages between data points[J]. Science, 2007, 0315(16): 972-976

[3] 张讲社，梁怡，徐宗本. 基于视觉系统的聚类算法[J]. 计算机学报，2001，24(5)：496-501

[4] Zhang Jiang-she, Liang Yin-wing. Improved possibilistic c-means clustering algorithms [J]. IEEE Trans on Fuzzy Systems, 2004, 12(2): 2009-217

[5] 于剑. 论模糊 c 均值算法的模糊指数[J]. 计算机学报，2003，26(8)：968-973

[6] 陈宗海，文锋，聂建斌，等. 基于节点生长 K-均值聚类算法的强化学习方法[J]. 计算机研究与发展，2006，43(4)：661-666

[7] Berchtolds, Bohm C, Keim D A, et al. A cost model for nearest neighbor search in high-dimensional data space [A] // Proceedings of the sixteenth ACM SIGMOD Symposium on Principles of database systems[C]. Tucson, Arizona: ACM Press, 1997

[8] Ertozi, Stenbachm, Kumarv. Finding clusters of different shapes and densities in noisy high-dimensional data[R]. Minnesota Department of Computer Science, University of Minnesota, 2002

[9] Ham J H, Lee D D, Saul L K. Learning high-dimensional correspondences from low dimensional manifolds [C] // Proc of ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining. Washington, 2003: 34-41

[10] Dash M, Liu J Y. Dimensionality reduction of unsupervised data [A] // Proc 9th IEEE Int Conf Tool with Artificial Intelligence [C]. 1997: 532-539

[11] Talavera L. Feature selection as a preprocessing step for hierarchical clustering[A] // Proc 16th Int Conf on Machine Learning [C]. 1999: 389-397

[12] Dy J G, Brodley C E. Feature subset selection and order identification for unsupervised learning [A] // Proc 17th Int Conf on Machine Learning[C]. 2000: 247-254

[13] Yang Yi-ming. An evaluation of statistical approaches to categorization[J]. Journal of Information Retrieval, 1999, 1(1/2): 67-88

[14] Huang Zhe-xue. Clustering large data sets with mixed numeric and categorical values [C] // Proc of PAKDD'97. Singapore: World Scientific, 1997: 21-35

[15] Huang Zhe-xue. Extensions to the K-means algorithm for clustering large data with categorical values [J]. Data Mining and Knowledge Discovery, 1998, 2(3): 283-304

(上接第 169 页)

出了一种海量规则模式匹配算法。

参 考 文 献

[1] Zhou J J, Yu K M. Tidset-based parallel FP-tree algorithm for the frequent pattern mining problem on PC clusters[J]. Lecture Notes in Computer Science, 2008, 50(3): 18-28

[2] Aref M M, Tayyri M A. Lana-Match algorithm: A parallel version of the Rete-Match[J]. Parallel Computing, 2008, 24(1): 763-775

[3] Gaudiot J L, Soha A. Data-driven parallel production systems [J]. IEEE Transactions on Software Engineering, 1990, 16(3): 281-293

[4] Wolfson O, Ozeri A. Parallel and distributed processing of rules by data-reduction[J]. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(3): 523-530

[5] Walzer K, Breddin T G. Matthias relative temporal constraints in the Rete algorithm for complex event detection[C] // Proceedings of the 2nd International Conference on Distributed Event-Based Systems. Rome, ITALY: DEBS 2008: 147-155

[6] Marsuzawa K. Parallel execution method of production systems with multiple worlds[C] // IEEE Int Workshop Tools Artif Intell Archit Lang Algorithms. SINGAPORE: IEEE, 1989: 339-344

[7] Forgy C L. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem[J]. Artificial Intelligence, 1982, 19(1): 17-37

[8] 张桂刚. 海量规则并行处理研究[D]. 武汉：武汉大学，2009