Journal of Universal Computer Science, vol. 13, no. 10 (2007), 1484-1497
submitted: 12/6/06, accepted: 24/10/06, appeared: 28/10/07 © J.UCS

A Model of Immune Gene Expression Programming for Rule Mining

Tao Zeng, Changjie Tang
(School of Computer, Sichuan University, China
zt1011@sina.com, tangchangjie@cs.scu.edu.cn)

Yong Xiang
(Chengdu Electromechanical College, China
xiangyong@cs.scu.edu.cn)

Peng Chen, Yintian Liu
(School of Computer, Sichuan University, China
chengpeng@cs.scu.edu.cn, liuyintian@cs.scu.edu.cn)

Abstract: Rule mining is an important issue in data mining. To address it, a novel **I**mmune **G**ene **E**xpression **P**rogramming (IGEP) model was proposed. Concepts of rule, gene, immune cell, and antibody were formalized. The dynamic evolution models and the corresponding recursive equations of immune cell, self, immune-tolerance were built. The novel key techniques of IGEP were presented. Experiment results showed that the new method has good stability, scalability and flexibility. It can discover traditional association rule, non-traditional rule including connective “OR” or “NOT”, and meta-rule of strong rule. Furthermore, it can perform well in constrained pattern mining.

Key Words: Data mining, Rule, Meta-rule, Evolutionary algorithm, Gene expression programming, Artifical immune system

Category: I.2.6, H.2.8, I.6.5, I.5.2, F.2.2

1 Introduction

Gene Expression Programming, Artificial Immune System, and Rule Mining are all hot research themes.

Gene **E**xpression **P**rogramming (GEP) [Ferreira 2001] is derived and improved from **G**enetic **P**rogramming (GP) [Banzhaf 1994]. It is a new technique to create programs, which can denote the learned models or discovered knowledge. GEP can represent and solve complex problem with simple code.

Artificial **I**mmune **S**ystem (AIS) [Jerne 1974, Burnet 1978, Forrest et al. 94, Castro et al. 1999, Castro et al. 2000, Dasgupta et al. 2003, Li et al. 2005] is a rapidly growing field of information processing based on immune inspired paradigms of nonlinear dynamics. It is expected that AIS, based on immunological principles, be good at modularity, autonomy, redundancy, adaptability, distribution, diversity and so on.

Rule Mining is an important data mining task since it generates a set of symbolic rules that describe each class or category in a natural way. Rule is easier to understand than other data mining model. So far fruitful research results for **A**ssociation **R**ule (AR) mining can be found in [Agrawal et al. 1993, Fu and Han 1995, Han and Kambr 2001, Yin and Han 2003].

However, complex data mining application requires refined and rich-semantic knowledge representation. For example, using traditional concepts and methods, it is difficult to describe and discover the rule or meta-rule in Example 1.

Example 1 Suppose that customers probably purchase "laptop" if age is "40-50", ***either*** title is "*prof.*", ***or*** address is ***not*** at "campus". To describe this fact, we need other new association rule in the form of

$$age(\text{“40-50”})\wedge(title(\text{“}prof.\text{”})\vee\neg address(\text{“campus”}))\rightarrow purchase(\text{“laptop”}) \quad (1)$$

$$age(x)\wedge(title(y)\vee\neg address(z))\rightarrow purchase(u) \quad (2)$$

where rule (2) is called meta-rule of rule (1) in this paper.

On the issue of mining the rule like Example 1, little related work can be retrieved except [Zuo et al. 2002]. In 2002, Zuo proposed an effective approach based on GEP [Zuo et al. 2002]. However, it can only mine single-dimensional predicate AR, without concerning multi-dimensional rule or meta-rule. Moreover, its flexibility and stability are not so good.

To overcome the above defects and mine more general rules, it is necessary to build a new model. GEP is strong on representing and discovering knowledge with simply linear strings while AIS has many advantages in evolution. To inherit and enhance their merits, we proposed a novel model "**I**mmune **G**ene **E**xpression **P**rogramming" (IGEP). IGEP is able to discover traditional AR, non-traditional rule including connective "OR" or "NOT", and meta-rule of strong rule. Furthermore, it can perform well in constrained pattern mining.

Main novel techniques of IGEP include: (a) distinctive structures of immune cell and antibody, based on which an antibody can represent 8 rules, (b) the **T**emplate-based **D**ual-**F**ormula **G**eneration **S**trategy (TDFGS) to guarantee quality of immune cell, (c) the Dynamic Self-Tolerance Strategy to eliminate both invalid and redundant immune cells, and (d) in "Affinity Computing", the rule **R**eduction **C**riterion (RC) that a strong rule is fine if and only if the contra-positive of it is strong too.

The rest of the paper is organized as follows. Section 2 describes the background and our motivation. Section 3 presents the IGEP Model, including some formal concepts and the framework. Section 4 gives the key techniques of IGEP. Section 5 shows our experiment results. Finally, Section 6 draws conclusions and gives directions of future work.

2 Background and Motivation

2.1 Gene Expression Programming

Gene **E**xpression **P**rogramming (GEP)[Ferreira 2001] is designed to solve complex problem with simple code. GEP is somewhat similar to **G**enetic **A**lgorithms (GA) [Mitchell 1996] or **G**enetic **P**rogramming (GP) [Banzhaf 1994]. The chromosome of GP is tree-formed structure directly, while that of GEP is linear string. So GP's genetic operations are designed to manipulate the tree forms of chromosomes. However, GEP's genetic operations are similar to but simpler than those in GA. Compared with its ancestors, GEP innovated in structure and method. It uses a very smart method to decode gene to a formula [Ferreira 2001, Zuo et al. 2002]. Figure 1 demonstrates the decoding process in GEP. As an example, if let "a", "b" and "c" represent atomic predicates "$age(x)$", "$title(x)$" and "$address(x)$" respectively, then the expression in Figure 1 can express the logic formula "$(age(x) \vee\ age(x)) \wedge (tile(x) \vee \neg address(x))$". In this way, the new model can represent and discover meta-rule.

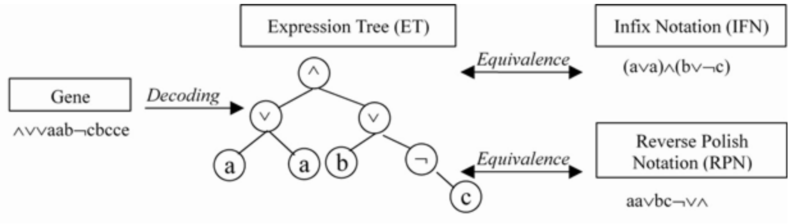

Figure 1: Decoding for gene in GEP

2.2 Artificial Immune System

The **B**iology **I**mmune **S**ystem (BIS) can defend the body against harmful diseases and infections. It is capable of recognizing virtually any foreign cell or molecule and eliminating it from the body. As a member of nature-inspired computing, AIS imitates BIS, aiming not only at a better understanding of the system, but also at solving engineering problems [Castro et al. 1999]. It is expected that AIS, based on immunological principles, be good at modularity, autonomy, redundancy, adaptability, distribution, diversity and so on. Although it has many features in common with neural networks, there are some differences: the immune system is more complex, more diverse, and it performs many different functions simultaneously.

With the development of applications, AIS gets more and more hot recently. The immune network theory [Jerne 1974], the clonal selection and affinity maturation algorithms [Burnet 1978], negative selection algorithm [Forrest et al. 94] and so on have greatly promoted the research of computer immune system. Moreover, there are many models and techniques for AIS based on different principles or representations. According to [Castro et al. 1999, Castro et al. 2000, Dasgupta et al. 2003], the main representations used include binary strings, real-valued vectors, strings from a finite alphabet, java objects and so on.

2.3 Motivation

GEP is strong on representing and discovering knowledge with simply linear strings. AIS has many advantages in evolution. It is natural to assume that embedding GEP in AIS will enhance the capability of both AIS and GEP. We call the new model as Immune Gene Expression Programming (IGEP).

3 IGEP Model

In this section, we will introduce some notations, concepts and our IGEP model. Notations and basic concepts on relational algebra are the same as those in [Han and Kambr 2001].

3.1 Concepts for Rule

Like [Yin and Han 2003], a ***literal*** p can be defined as an attribute-value pair, taking the form of (A_i, v), in which A_i is an attribute and v a value. A tuple t ***satisfies*** a literal p = (A_i, v) if and only if $t_i = v$, where t_i is the value of the i^{th} attribute of t.

In addition, ϑ_p denotes the atomic first-order predicate that corresponds to literal p, which means that the value of attribute A_i is v. Let ζ be a literal set and we write the atomic predicate set $\zeta^{\vartheta} = \{x \mid x = \vartheta_y, \forall y \in \zeta\}$.

The definition of rule in this paper, distinguished from [Fu and Han 1995, Yin and Han 2003], is as follows.

Definition 1. Let ζ be a literal set, OP=$\{\neg, \wedge, \vee\}$ be a connective set,X,$Y \subset \zeta^{\vartheta}$, X, $Y \neq \phi$, and $X \cap Y = \phi$. A ***rule*** r is an expression in the form of $P \to Q$ where

- P, called ***antecedent,*** is a well-formed first-order logic formula composed of atomic formulas in $X$and connectives in OP.
- Q, called ***consequent,*** is a well-formed first-order logic formula composed of atomic formulas in $Y$and connectives in OP.

– If $\forall\ p = (A_i, v) \in \zeta$, the v in p is replaced with a variable, then the new rule is the ***meta-rule*** of the origin one.

Let $f(p,\, t)$ denote whether a tuple t satisfies a literal p.

$$f(p,\, t) = \begin{cases} true & \text{if } t \text{ satisfies } p \\ false & \text{otherwise} \end{cases} \tag{3}$$

Given $L \in \{P,\, Q,\, P \wedge Q\}$ and t be a tuple in relation, we write the notation $S(L, t)$ for the Boolean formula substituted for L, where, for each literal p corresponding to the atomic first-order predicate in L, we replace all ϑ_p with $f(p,t)$.

Definition 2. A tuple t ***support*** $L \in \{P,\, Q,\, P \wedge Q\}$ if and only if the evaluation result of $S(L, t)$ is true; otherwise, ***not support***.

Let $\rho(L|D)$ denote the number of records that support $L \in \{P,\, Q,\, P \wedge Q\}$ on a data set D. $\#(D)$ is the total number of records in D. Then the ***support degree*** $supp(r|D)$ and the ***confidence degree*** $conf(r|D)$ of a rule r can be valuated as follows.

$$supp(r|D) = \tfrac{\rho(P \wedge Q|D)}{\#(D)}) \tag{4}$$

$$conf(r|D) = \tfrac{\rho(P \wedge Q|D)}{\rho(P|D)} \tag{5}$$

Let min_conf, $min\_sup \in [0, 1]$. r is ***strong*** if and only if $supp(r \mid D) \geq min\_sup$ and $conf(r \mid D) \geq min\_conf$ like [Han and Kambr 2001].

It is easy to prove that the rule referred to in Definition 1 is equivalent to the traditional AR if and only if (a) $OP = \{\wedge\}$, (b) each of atomic predicates in it occurs only once, and (c) the order of atomic predicates in it is not considered. Thus the rule referred to in this paper is more general than traditional AR.

Lemma 3. *If FS=$\{A, B\}$ be the set composed of antecedent and consequent of a rule, then FS can be used to construct 8 rules, which can be grouped as 4 pairs. Each pair of these 4 pairs are equivalent in logic each other.*

Proof. we can construct the following 8 rules: a) $A \rightarrow B$, b) $\neg B \rightarrow \neg A$, c) $B \rightarrow A$, d) $\neg A \rightarrow \neg B$, e) $\neg A \rightarrow B$, f) $\neg B \rightarrow A$, g) $A \rightarrow \neg B$, and h) $B \rightarrow \neg A$. In them, a) and b), c) and d), e) and f), g) and h) are the contra-positive each other respectively. Since the contra-positive is equivalent to the original statement, two statements in pair are equivalent each other.

Lemma 4. *Let FS=$\{A, B\}$ be the set of antecedent and consequent of a rule, and a relation instance D. If $\rho(A|D)$,$\rho(B|D)$,$\rho(A \wedge B|D)$ and $\#(D)$ were given, then all of support degree and confidence degree for 8 rules constructed by FS can be evaluated.*

Proof. Figure 2 shows the support space for rule. Because in our system, arbitrary tuple can either support a rule or not, we can compute the following value: 1) $\rho(\neg A|D) = \#(D)$ - $\rho(A|D)$, 2) $\rho(\neg B|D) = \#(D)$ - $\rho(B|D)$, 3) $\rho(A \wedge \neg B|D) = \rho(A|D)$ - $\rho(A \wedge B|D)$, 4) $\rho(\neg A \wedge B|D) = \rho(B|D)$ - $\rho(A \wedge B|D)$, 5) $\rho(\neg A \wedge \neg B|D) = \#(D)$ - $\rho(A|D)$ - $\rho(B|D)$ + $\rho(A \wedge B|D)$. Using these values, we can evaluate support degrees and confidence degrees for these rules by Equation (4) and (5).

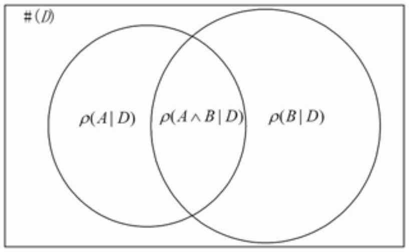

Figure 2: Support space for rule

3.2 Concepts for IGEP

The gene in IGEP can represent complex expression with simple structure like GEP [Ferreira 2001, Zuo et al. 2002]. The formal description is as follows.

Definition 5. Let T be the terminal set and OP be the operator set. A ***Gene*** is a linear string composed of the elements in T and OP.

In this paper, T=ζ^{ϑ}, and OP can be one element of $2^{\{\neg,\wedge,\vee\}}$ - $\{\phi\}$.

Definition 6. The ***Decoding*** is a procedure where a gene can be decoded into a well-formed expression tree or string.

Immune cell and antibody are very important for AIS. In general, antigen is corresponding to the problem to be solved and antibody to the solution for it.

For rule mining problem, records in data set can be antigen and rules can be antibody. The formal descriptions of immune cell and antibody are as follows.

Definition 7. An immune cell, ***BCell***, is a 3-tuple (C, F, η) where

- $C = (g_A, g_B)$ is a 2-tuple, called ***Chromosome***, where g_A and g_B are genes.
- $F = (e_A, e_B)$ is a 2-tuple, called ***dual-formula***, which were decoded from genes in C respectively.

- $\eta \in${-1, 0, 1, 2} is the state value of BCell, where -1, 0, 1 and 2 indicate cell is dead, immature, mature and memorized respectively.

Definition 8. An ***antibody*** is a 3-tuple, (F, S, I), where

- F comes from the immune cell that produces it.
- $S = (s_A, s_B)$ is a 2-tuple, where s_A and s_B are the substitution formulas for those in F respectively by atomic predicates derived from literals.
- $I = (p_A, p_B, p_{AB}, p_{total})$ is a 4-tuple, which stores affinity information. In I, p_A, p_B, p_{AB} and p_{total} are the support numbers of s_A, s_B and $s_A \wedge s_B$ and the total number of records that were matched respectively.

Theorem 9. *An antibody can represent and evaluate 8 rules.*

Proof. Let Ab denote an antibody, and $A{=}Ab.S.s_A$, $B{=}Ab.S.s_B$. Then by Lemma 3 an antibody can represent 8 rules by using $\{A, B\}$. After affinity maturation, there are $\rho(A|D){=}Ab.I.p_A$, $\rho(B|D){=}Ab.I.p_B$, $\rho(A{\wedge}B|D){=}Ab.I.p_{AB}$, and $\#(D){=}Ab.I.p_{total}$. We can evaluate these 8 rules by Lemma 4.

It shows our antibody is good at representation and discovery of rules.

3.3 IGEP Framework

Since GEP is strong on representing and discovering knowledge with simply linear strings while AIS has many advantages in evolution, we propose the new method as **I**mmune **G**ene **E**xpression **P**rogramming (IGEP).

The framework of IGEP is somewhat similar to the hybrid of clonal selection principle [Burnet 1978] and negative selection algorithm [Forrest et al. 94]. In contrast to other models [Dasgupta et al. 2003], IGEP has distinctive structures of immune cell and antibody, and other novel key techniques. The flowchart of IGEP is described in Figure 3.

4 Key Techniques of IGEP

4.1 Dual-Formula Generation Strategy for Immune Cell Generation

It is possible to focus on mining some rules with special form or those who represent the correlation of special attributes or items. For example, we want only to mine rules in which each literal occurs *only once* such as "a∧(b∨¬c) → d". However, traditional GEP may randomly generate formulas like "(a∨a)∧(b∨¬c)" too. So the rule we do not want can be also constructed. Because the cost of removing fault antibody will be relatively high, we proposed the **T**emplate-based

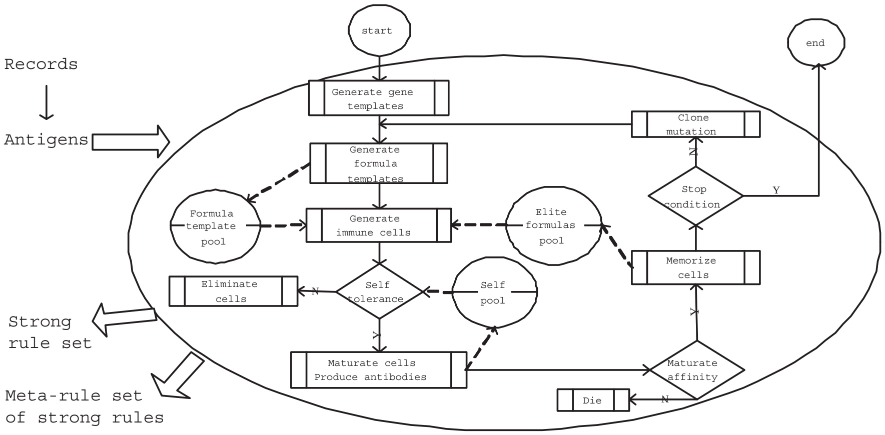

Figure 3: The flowchart of IGEP

Dual-**F**ormula **G**eneration **S**trategy (TDFGS). It is via TDFGS that IGEP can always generate valid dual-formulas according to system requirements.

Given a literal set ζ and the atomic predicate set ζ^{ϑ}, main steps of TDFGS are:

Step 1: Let terminal set $T = \{\#\}$, function set OP, call "Generate gene templates" to generate genes and decode them into expression strings, called **F**ormula **T**emplates **(FTemp).**

Step 2: Take two FTemps ft_A and ft_B from FTemp pool according to requirements for the form of dual-formula. If lost, then do nothing and return NULL; else success, (ft_A, ft_B) is selected.

Step 3: Suppose $W \subseteq \zeta^{\vartheta}$, and take predicates in W to fill "#" in ft_A and ft_B where the attribute or items can be filtered and controlled. So dual-formula is generated according to system requirements.

The functions of TDFGS are as follows.

- It guarantees each of dual-formula of BCell can construct valid rules.

- It is easy to inject vaccine into the AIS of IGEP. Filter out or select formula templates by certain pattern and we can concentrate on those rules that we just want but not face all possible rules.

- In Step 3, the attributes or items in rules can be selected and we can focus on discovering the correlation between certain attributes or items.

4.2 Dynamic Immune Tolerance Strategy

The part of self-tolerance in IGEP develops from negative select algorithm [Forrest et al. 94] and looks like that in [Li et al. 2005]. But there are many differences from them. The formal descriptions of dynamic immune tolerance strategy of IGEP are as follows.

$$BCSet_{mature}(t){=}BCSet_{immature}(t)\text{-}\ BCSet_{dead}(t) \tag{6}$$

$$BCSet_{dead}(t){=}BCSet_{immature}(t)\cap(\mathit{SelfBCs}(t\text{-}1)\cup \mathit{SelfBCs}_{equivalent}(t\text{-}1)) \tag{7}$$

$$\mathit{SelfBCs}(t) = \begin{cases} \{x|x \textit{ is the BCell involved in vaccine}\} & t=0 \\ \mathit{SelfBCs}(t-1)\cup BCSet_{immature}(t) & t\geq 1 \end{cases} \tag{8}$$

where

$$BCSet_{mature}(t){=}\{x|x \textit{ is the mature BCell generated at generation } t\} \tag{9}$$

$$BCSet_{immature}(t){=}\{x|x \textit{ is the immature BCell generated at generation } t\} \tag{10}$$

$$BCSet_{dead}(t) = \{x|x \textit{ is the BCell eliminated at generation } t\} \tag{11}$$

$$\mathit{SelfBCs}(t){=}\{x|x \textit{ is the BCell involved in self at generation } t\} \tag{12}$$

$$\mathit{SelfBCs}_{equivalent}(t){=}\{x|x \in BCs_{equivalent}(bc),\ bc{\in}\mathit{SelfBCs}(t)\} \tag{13}$$

$$\begin{aligned} BCs_{equivalent}(bc){=}\{x|x \textit{ is the BCell, } x.F \textit{ is one of } (e_B, e_A), (\neg e_A, e_B),\\ (e_B, \neg e_A), (e_A, \neg e_B), (\neg e_B, e_A), (\neg e_A, \neg e_B\ ), \textit{ and } (\neg e_B, \neg e_A),\\ \textit{where } bc \textit{ is a BCell, } bc.F{=}(e_A{,}e_B)\ \} \end{aligned} \tag{14}$$

Equation (6) and (7) depict the dynamic immune tolerance strategy, while Equation (8) describes the dynamic evolution of self. It is because there is *Self-*$BCs_{equivalent}(t\text{-}1)$ in Equation (7) that IGEP can avoid generating cells with redundant representation.

The functions of our dynamic tolerance strategy are as follows.

- Avoid generating redundant cells that are equivalent to represent rule.
- Avoid generating fault cells that cannot represent valid rules.
- Be able to inject vaccine.

4.3 Affinity Computing

In course of affinity maturation, for each antibody, its affinity information for all records (antigens) will be computed. After affinity maturation, there are $\rho(Ab.S.s_A|D) = Ab.I.p_A$, $\rho(Ab.S.s_B|D) = Ab.I.p_B$, $\rho(Ab.S.s_A \wedge Ab.S.s_B|D) = Ab.I.p_{AB}$, and $\#(D) = Ab.I.p_{total}$. According to Theorem 9, Equation (4) and (5), we can scan database once but evaluate 8 times more rules than antibodies. Then system will be able to mine strong rules for output.

Additionally, IGEP can reduce result set based on the heuristic **R**eduction **C**riterion (RC) that a strong rule is fine if and only if the contra-positive of it is strong too, for the statement and contra-positive is logically equivalent.

5 Experimental Evaluation

5.1 Experimental Setup

Our test platform is as follows. CPU: AMD XP 2500+, memory: 1GB, hard disk: 160GB, OS: MS Windows XP Pro. SP2, compiler: JDK1.5.03. All of 3 data sets we used in our experiments come from UCI Machine Learning Repository[1] .

The data sets are Tic-Tac-Toe Endgame database (*ttt*) with 9 attributes plus 1 class column and 958 rows, Car Evaluation Database (*car*) with 7 attributes and 1728 rows, and Contraceptive Method Choice(*cmc*) with 10 attributes and 1473 rows. Table 1 gives us notation definitions for this section.

Additionally, we call a rule as ***h-rule*** if and only if the number of attributes involved in it is h, and those attributes occur only once in it. As an example, the rule (1) in Example 1 is a 4-rule. In our experiments, the objective to mine is h-rule but not general rule, for h-rule not only has smaller solution space but also is more extractive and heuristic for us to understand. In fact, because there are more constraints to h-rule than general rule, it needs more complex algorithms to mine h-rule than general rule.

5.2 Mining Rule

We take the mining results via Apriori algorithm [Agrawal and Srikant 1994] as a baseline to verify IGEP. In order to utilize Apriori algorithm to mine multi-dimensional AR, we always preprocess data sets for it in the following way. For each value of attribute in a data set d, we add a string of its attribute in front of it to construct a new value, whose type become string, then store it into a new data set d'. After preprocessing, in d', original equal values in different attributes in d became unequal. Potential value-collisions between dimensions

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

have been eliminated before Apriori runs on d'. So we can take such record sets as transaction set to mine multi-dimensional AR via Apriori.

In Table 2, extensional tests showed that 1) our algorithm is stable, 2) the efficiency of our heuristic reducing criterion RC is notable by comparison between No 4 and 5 or 6 and 7, 3) the capability of generating new immune cells is strong, and 4) the function of vaccine is sound and effective. As an example, a 5-rule from results of No.9 in Table 2 is as follows.

$$\mathrm{D}_7(1)\wedge\mathrm{D}_8(4)\wedge(\mathrm{D}_6(1)\vee\mathrm{D}_2(1))\rightarrow\neg\mathrm{D}_3(2)\ \text{supp=14.53\% conf=99.53\%} \tag{15}$$

$$\mathrm{D}_3(2)\rightarrow\neg(\ \mathrm{D}_7(1)\wedge\mathrm{D}_8(4)\wedge(\mathrm{D}_6(1)\vee\mathrm{D}_2(1)))\ \text{supp=12.02\% conf =99.44\%} \tag{16}$$

$$\mathrm{D}_7(x_7)\wedge\mathrm{D}_8(x_8)\wedge(\mathrm{D}_6(x_6)\vee\mathrm{D}_2(x_2))\rightarrow\neg\mathrm{D}_3(x_3) \tag{17}$$

where D_i(c) denotes the value of i^{th} attribute is c.

Rule (15) and (16) can be reduce to a 5-rule, because they are equivalent each other in logic. Rule (17) is the meta-rule of strong 5-rule (15).

Table 1: More notations for section 5

Notation	Definition
cellnum	The maximum of BCells per generation
PO	Whether to consider the order of atomic predicates in a rule
NC	Number of cells
SR	Number of strong rules
MR	Number of meta-rules
SAR	Number of strong traditional multi-dimensional ARs
ECN	Number of cells eliminated by self tolerance

5.3 Scalability Study

Firstly, we study on time wasted by main processes of IGEP. Figure 4 showed information about time wasted of someone generation on different data sets. It indicated 1) for each generation, time wasted by processes of IGEP was relatively stable, and 2) the process of "Maturate affinity" consumed most time while "Generate BCell" took less time. Thus, based on 2) above, it is valuable to spend more time on improving the quality of BCell generated. We infer our IGEP, due to having TDFGS and dynamic immune tolerance strategy, be stronger than the method only based on traditional GEP.

Secondly, we evaluate scalability of IGEP on different data sets in the following way. Basic parameters are fixed and each data set is divided to 4 segments. For line "incremental", data sets, built on these 4 segments incrementally, were

Table 2: Results for minig h-rule *min_supp*=5.0% *min_conf*=98.5% *cellnum*=20

No.	Data	h	PO	OP	RC	NC	ECN	IGEP		Apriori
								MR	SR	SAR
1	*ttt*	2 to 10	No	$\{\wedge\}$	No	28501	77846	10	12	12
2	*car*	2 to 7	No	$\{\wedge\}$	No	966	125247	12	40	40
3	*cmc*	2 to 10	No	$\{\wedge\}$	No	28501	78132	126	228	228
4	*cmc*	3	Yes	$\{\neg, \wedge, \vee\}$	No	5760	30966	10412	316292	Disable
5	*cmc*	3	Yes	$\{\neg, \wedge, \vee\}$	Yes	5760	58411	1424/2	1960/2	Disable
6	*cmc*	4	Yes	$\{\neg, \wedge, \vee\}$	No	10000	46	19998	1334128	Disable
7	*cmc*	4	Yes	$\{\neg, \wedge, \vee\}$	Yes	10000	64	4314/2	13592/2	Disable
8	*car*	2 to 7	No	$\{\neg, \wedge, \vee\}$	Yes	10000	3250	3326/2	412784/2	Disable
9	*cmc*	2 to 6	Yes	$\{\neg, \wedge, \vee\}$	Yes	10000	878	4096/2	12862/2	Disable
10	*car*	5	Yes	$\{\neg, \wedge, \vee\}$	No	2520	86314	24	336	Disable

Notes:

- All of data sets used by Apriori algorithm had been preprocessed and their results are presented as antitheses to those of IGEP.
- The numbers of independent MR and SR are the original values divided by 2 if RC was used.
- For No. 1 to 5 and 10, MR and SR are stable while the others can change within a certain range in different tests.
- In No. 9, attributes were restricted to 2^{nd}, 3^{rd}, 4^{th}, 6^{th}, 7^{th} and 8^{th}.
- In No. 10, the dual-formula template was ("#", "(#∨¬#)∧(#∨#)").

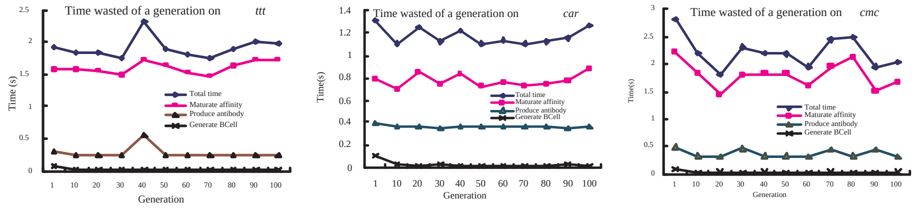

Figure 4: Time wasted study on different data sets for mining 4-rule, *cellnum*=20,PO = No, and $OP = \{\neg, \wedge, \vee\}$. The data set is (a) *ttt*, (b) *car*, and (c) *cmc* respectively.

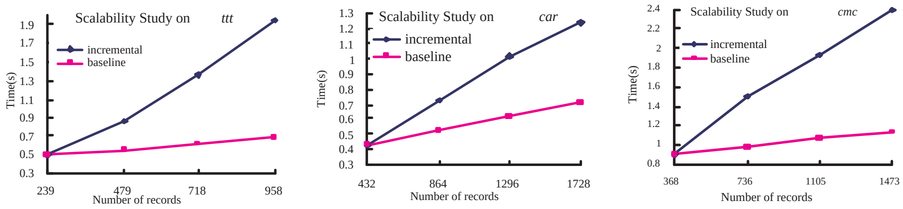

Figure 5: Relationship between average running time per generation and the number of records taken from different data sets incrementally for mining 4-rule, *cellnum*=20,PO =No, and OP= $\{\neg, \wedge, \vee\}$. The data set is (a) *ttt*, (b) *car*, and (c) *cmc* respectively.

mined 4 times respectively. For "baseline", data sets come from the first segment d, double of d, triple of d, and quadruple of d respectively.

Figure 5 described results about scalability study on *ttt*, *car*, and *cmc*. It showed the average running time per generation depends on the number of unique records in data set, and increases approximately linearly with the number of records on these data sets. Table 3 gives the comparison between IGEP, PAGEP in [Zuo et al. 2002], and Apriori[Agrawal and Srikant 1994].

Table 3: Comparison between IGEP, PAGEP, and Apriori

Function	**IGEP**	**PAGEP**	**Apriori**
Mining traditional association rule	Yes	Yes	Yes
Mining rule including connective "OR" or "NOT"	Yes	Yes	No
Mining meta-rule of strong rule	Yes	No	No
Mining rule complying with constrained pattern	Yes	No	No
Mining rule related to constrained attributes	Yes	No	No

6 Conclusions and Future Work

We proposed the IGEP model for rule mining, formalized basic concepts and presented some novel key techniques of IGEP. Experiment results showed that the new method has good stability, scalability and flexibility. It can discover traditional association rule, non-traditional rule including connective "OR" or "NOT", and meta-rule of strong rule. Furthermore, it also can perform well in constrained pattern mining.

Our future works will be focused on improvement of performance, discovery of rule on data streams, and application of text mining or web log mining.

Acknowledgements

This paper has been supported by the National Science Foundation of China under Grant Nos. 60473071 and 90409007.

References

[Agrawal et al. 1993] Agrawal R., Imiclinski T., Swami A.: "Database mining: A performance perspective"; IEEE Trans Knowledge and Data Enginnering, 5(1993), 914-925

[Agrawal and Srikant 1994] Agrawal R., Srikant R.: "Fast Algorithm for Mining Association Rules" ; "Proceeding 1994 International Conference Very Large Data Bases (VLDB'94)", (1994)

[Banzhaf 1994] Banzhaf W.: "Genotype-phenotype-mapping and Neutral variation - A Case Study in Genetic Programming"; Parallel Problem Solving from Nature III, LNCS, 866 (1994)

[Burnet 1978] Burnet F. M.: "Clonal Selection and After"; "Theoretical Immunology" (Bell G. I., Perelson A. S., Pimbley G. H., eds.), Marcel Dekker Inc, New York (1978), 63-85

[Castro et al. 1999] De Castro L. N., Von Zuben F. J.:"Artificial Immune Systems: Part I-Basic Theory and Applications" ; Technical Report, TR-DCA Ol/99, 12 (1999)

[Castro et al. 2000] DE Castro L. N., Von Zuben F. J.:"Artificial Immune Systems: Part II-A Survey of Applications"; Tech Rep-RT DCA, 2(2000)

[Dasgupta et al. 2003] Dasgupta D., Ji Z., Gonzalez F.: "Artificial Immune System (AIS) Research in the Last Five Years"; Evolutionary Computation, 2003. CEC 03. The 2003 Congress, (2003), 123-130

[Ferreira 2001] Ferreira C.: "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems"; Complex Systems, 13, 2(2001), 87-129

[Forrest et al. 94] Forrest S., Perelson A. S., et al.: "Self-Nonself Discrimination in a Computer"; "Proceedings of IEEE Svmposiimi on Research in Secwitv and Privacy", 1994

[Fu and Han 1995] Fu Y., Han J.: "Meta-rule-guided Mining of Association Rules in Relational Databases"; KDOOD'95, Singapore, (1995), 39-46

[Jerne 1974] Jerne N. K.: "Towards a network theory of the immune system Annals of Immunology"; 125, C(1973), 373-389

[Han and Kambr 2001] Jiawei Han, Micheline Kambr: "Data Mining-Concepts and Techniques"; Higher Education Press, Bejing (2001)

[Li et al. 2005] Tao Li, Xiaojie Liu, and Hongbin Li: "A New Model for Dynamic Intrusion Detection"; CANS 2005, LNCS, 3810 (2005), 72-84

[Mitchell 1996] M. Mitchell: "An Introduction to Genetic Algorithms"; MIT Press, 1996

[Silberschatz et al. 2001] Silberschatz, Korth: "Databse System Concepts"; Fourth Edition, McGraw-Hill Computer Science Series, 2001

[Yin and Han 2003] Xiaoxin Yin, Jiawei Han: "CPAR: Classification Based on Predictive Association Rules"; "Proc. SIAM Int. Conf. on Data Mining (SDM'03)", (2003), 331-335

[Zuo et al. 2002] Jie Zuo, Changjie Tang, et al.: "Mining Predicate Association Rule by Gene Expression Programming"; WAIM 2002, LNCS, 2419 (2002), 92-103