Association Rule Mining Using Genetic Network Programming

Kaoru Shimada
Graduate School of Information, Production and Systems, Waseda University, Japan

Kotaro Hirasawa
Graduate School of Information Production and Systems, Waseda University, Japan

Takayuki Furutsuki
Graduate School of Information Production and Systems, Waseda University, Japan

Abstract

A method of association rule mining using Genetic Network Programming (GNP) is proposed to improve the performance of rule extraction. The proposed system can evolve itself by an evolutionary method and measures the significance of the association via the chi-squared test using GNP. In this paper, we describe the algorithms capable of finding the important association rules and present some experimental results. GNP examines the attribute values of database tuples using judgement nodes and calculates the measurements of association rules using processing nodes. The proposed method measures the significance of associations via the chi-squared test for correlation used in classical statistics, where GNP evolves itself using it as a part of the fitness value. Accordingly, the algorithms can extract the important association rules efficiently. Extracted association rules are stored in a pool all together through generations in order to find new important rules. Therefore, the proposed method is fundamentally different from all other evolutionary methods in its evolutionary way.

Keywords

Evolutionary Computation, Genetic Network Programming, Data Mining, Association Rule

1 Introduction

Association rule mining is the discovery of association relationships or correlations among a set of attributes in a database [1]. Association rule in the form of ‘ If X then Y ’ is interpreted as ‘ database tuples satisfying that X (antecedent) are likely to satisfy Y (consequent) ’. Association rules are widely used in marketing, decision making, and business management. Agrawal et al. have built a support-confidence framework for mining association rules from databases [2]. This model measures the uncertainty of an association rule with two factors: support and confidence. However, the measure is not adequate for modeling all uncertainties of association rules. For instance, the measurement does not provide a test for capturing the correlation of two itemsets. In order to improve this framework, some measurements on the support and confidence of association rules, such as chi-squared test model have been recently proposed by Brin et al [3]. The chi-squared test method measures the significance of associations via the chi-squared test for correlation used in classical statistics. However, it is difficult to use in case that the number of items included in association rules is increased.

Genetic Network Programming (GNP) [4, 5, 6] is a kind of evolutionary methods, which evolves arbitrary directed graph programs and includes judgement nodes and processing nodes in the network. GNP is useful because it can form not only the optimal structure effectively, but it also avoid the premature convergence. In this paper, we describe the algorithms capable of finding the important association rules using GNP to improve the performance of rule extraction. Attributes (items) in database correspond to judgement nodes in GNP, respectively. We are able to represent the connection of nodes as association rules and nodes are reused and shared with some other association rules because of GNP ’s feature. This method measures the support, confidence and significance of associations via the chi-squared test for correlation used in classical statistics using GNP. GNP evolves itself by an evolutionary method using chi-squared values as a part of the fitness value. Using genetic operation of GNP, we are able to obtain candidates of important association rules. Accordingly, the algorithms can extract the important association rules efficiently. In addition, extracted association rules are stored all together through generations and GNP evolves in order to find new interesting rules. Therefore, the method is fundamentally different from all other evolutionary algorithms.

2 Genetic Network Programming

In this section, the outline of Genetic Network Programming (GNP) [4, 5, 6] is explained. GNP is one of the evolutionary optimization techniques, which uses network structures as solutions. The basic structure of GNP is shown in Fig.1. GNP is composed of two kind of nodes: judgement node and processing node. Judgement nodes correspond nearly to elementary functions of Genetic Programming (GP). Judgement nodes are the set of J_1, J_2,..., J_m, which work as *if-then* type decision making functions. On the other hand, processing nodes are the set of P_1, P_2,..., P_n, which work as some kind of action/processing functions. The practical roles of these nodes are predefined and stored in the library by supervisors. Once GNP is booted up, the execution starts from Start node, then the next node to be executed is determined according to the connection from the current activated node.

The genotype expression of GNP node is shown in Fig.2. This describes the gene of node i, then the set of these genes represents the genotype of GNP individuals. NT_i describes the node type, $NT_i = 0$ when the node i is judgement node, $NT_i = 1$ when the node i is processing node. ID_i is an identification number, for example, $NT_i = 0$ and $ID_i = 1$ mean node i is J_1. C_{i1}, C_{i2},..., denote the nodes which are connected from node i firstly, secondly,..., and so on depending on the arguments of node i. d_i and d_{ij} are the delay time. They are the time required to execute the processing of node i and delay time from node i to node C_{ij}, respectively. All programs in a population have the same number of nodes, and the nodes with the same node number have the same function, respectively. The following genetic operators are used in GNP. Mutation operator affects one individual. All the connections of each node are changed randomly by mutation rate of P_m. Crossover operator affects two parent individuals. All the connections of the uniformly selected corresponding nodes in two parents are swapped each other by crossover rate P_c. GNP evolves the fixed number of nodes and these operators only change the connections among the nodes.

3 Association Rules

The following is a formal statement of the problem of mining association rules [1, 2] . Let $I = \{i_1, i_2, \ldots, i_l\}$ be a set of literals, called items or attributes. Let D be a set of transactions, where each

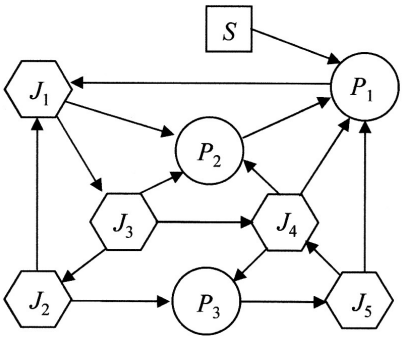

Figure 1: The basic structure of GNP individual

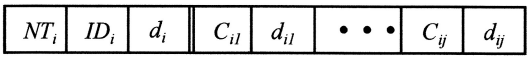

Figure 2: Gene structure of GNP (node i)

transaction T is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier, called TID. We say that a transaction T contains X, a set of some items in I, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. X is called the antecedent and Y is called the consequent of the rule. In general, a set of items is called an itemset. Each itemset has an associated measure of statistical significance called *support*. If the fraction of transactions in D containing X equals s, then we say that $support(X) = s$. The rule $X \Rightarrow Y$ has a measure of its strength called *confidence* defined as the ratio of $support(X \cup Y)/support(X)$. An example is shown below using Table 1. Let item universe be $I = \{A, B, C, D\}$ and transaction universe be $TID = \{1, 2, 3, 4\}$. In order to extend our research not only to market baskets problems but also to others, we indicate the items of the transaction by a 1 as shown in Table 1. In Table 1, itemset $\{A, C\}$ occurs in two transactions of $TID = 1$ and $TID = 3$. So, its frequency is 2, therefore, its support, that is, $support((A = 1) \wedge (C = 1))$ becomes 0.5. Itemset $\{A, C, D\}$ occurs in the transaction of $TID = 3$. Its frequency is 1, and its support, i.e., $support((A = 1) \wedge (C = 1) \wedge (D = 1))$ becomes 0.25. Therefore, $support((A = 1) \wedge (C = 1) \Rightarrow (D = 1)) = 0.25$, and $confidence((A=1) \wedge (C=1) \Rightarrow (D=1)) = 0.5$.

Table 1: An example of database

TID	A	B	C	D
1	1	0	1	0
2	0	1	1	1
3	1	1	1	1
4	0	1	0	1

Calculation of χ^2 value of the rule $X \Rightarrow Y$ is described as follows [3]. Let $support(X) = x$, $support(Y) = y$, $support(X \wedge Y) = z$ and the number of database tuples equals N. If events X and Y are independent then $support(X \wedge Y) = xy$. Table 2 is the contingency of X and Y : the upper parts are expectation values under the assumption of independence, and the lower parts are observational. Now, let E denote the value of expectation value under the assumption of independence and O the value of observational. Then the chi-squared statistic is defined as follows:

$$\chi^2 = \sum_{AllCells} \frac{(O-E)^2}{E} \tag{1}$$

We calculate χ^2 using x, y, z and N of Table 2.

$$\chi^2 = \frac{N(z-xy)^2}{xy(1-x)(1-y)} \tag{2}$$

This has 1 degree of freedom. If it is higher than a cutoff value (3.84 at the 95% significance level, or 6.63 at the 99% significance level), we reject the independence assumption.

Table 2: The contingency of X and Y

	Y	$\neg Y$	$\sum_{row}$
X	Nxy Nz	$N(x-xy)$ $N(x-z)$	Nx
$\neg X$	$N(y-xy)$ $N(y-z)$	$N(1-x-y+xy)$ $N(1-x-y+z)$	$N(1-x)$
$\sum_{col}$	Ny	$N(1-y)$	N

4 Association Rule Mining Using GNP

In this section, a method of association rule mining using GNP is proposed. Let A_i, B_i be attributes (items) in a database and its value is 1 or 0. The method extracts the association rule as follows:
$(A_j = 1) \wedge \cdots \wedge (A_k = 1) \Rightarrow (B_m = 1) \wedge \cdots \wedge (B_n = 1)$
(briefly, $A_j \wedge \cdots \wedge A_k \Rightarrow B_m \wedge \cdots \wedge B_n$).

4.1 GNP for Association Rule Mining

Attributes in the database correspond to judgement nodes in GNP, respectively. We are able to represent the connection of nodes as association rules. GNP examines the attribute values of database tuples using judgement nodes and calculates the measurements of association rules using processing nodes. The measurements include *support* and *confidence*. Judgement node determines the next node by a judgement result (Yes/No). Fig.3 shows a basic structure of GNP. P_1 is a processing node and is a starting point of association rules. Each Processing node have an inherent numeric order (P_1, P_2,..., P_n) and basically are connected from a judgement node. Yes-side of judgement node is connected to another judgement node. Judgement nodes can be reused and shared with some other association rules because of GNP 's feature. No-side of judgement node is connected to the next numbered processing node. We now demonstrate this using an example. In Table 1, the tuple $TID = 1$ satisfies $A = 1$ and $B \neq 1$, therefore the moving is from P_1 to P_2 in Fig. 3. If the examination of the connection from the stating point P_n is ended, then GNP examines $TID = 2$ likewise. Thus, all tuples in database will be examined. The total number of tuples moving to Yes-side at each judgement nodes are calculated for every processing node, which is a starting point for calculating association rules. All GNP individuals are searched parallel at the same time. If Yes-side connection of judgement nodes continue and the number of their judgement nodes becomes a cutoff value (maximum number of attributes in extracted association rules), then Yes-side connection is transferred to the next processing node obligatorily.

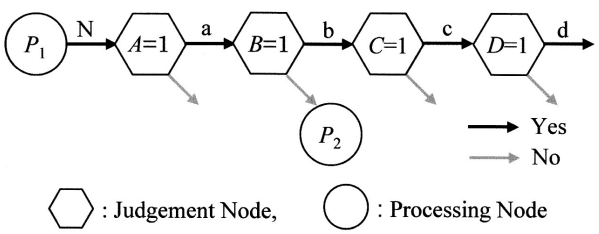

Figure 3: GNP for association rule mining

4.2 Extraction of Association Rules

In Fig.3, N is the number of total tuples, and a, b, c and d are the numbers of tuples moving to Yes-side at each Judgement node. Table 3 shows the measurements of association rules. The proposed method measures the significance of associations via the chi-squared test for correlation used in classical statistics. For example, if we change the connection of P_1 node from '$A = 1$' node to '$B = 1$' node (judgement node) in Fig.3, then we are able to calculate the support of B, $B \wedge C$ and $B \wedge C \wedge D$ in the next examination. As a result, we obtain chi-squared statistics and repeat this like a chain operation. We can define important association rules as the rules which satisfy the following:

$$\chi^2 > 6.63 \tag{3}$$

$$support \geq sup_{min} \tag{4}$$

sup_{min} is the threshold minimal support given by supervisors. The extracted important association rules are stored in a pool all together through generations in order to find new important rules. When an important rule is extracted, the overlap of the attributes is checked and it is also checked whether an important rule is new or not. If the rule is new, it is stored in the pool with its *support*, *confidence* and χ^2. Therefore, the method is fundamentally different from all other evolutionary algorithms.

Table 3: Association rules

association rules	support	confidence
$A \Rightarrow B$	b/N	b/a
$A \Rightarrow B \wedge C$	c/N	c/a
$A \Rightarrow B \wedge C \wedge D$	d/N	d/a
$A \wedge B \Rightarrow C$	c/N	c/b
$A \wedge B \Rightarrow C \wedge D$	d/N	d/b
$A \wedge B \wedge C \Rightarrow D$	d/N	d/c

4.3 Genetic Operators

Fitness evaluation function of GNP is defined as

$$F = \sum_{i \in I} \{ {\chi_i}^2 + 10(n(i_{ante}) - 1) + 10(n(i_{con}) - 1) + \alpha_{i_{new}} \} \quad (5)$$

The components are as follows:
I : a set of the number of important association rules which satisfy (3) and (4) in a GNP (individual)
$n(i_{ante})$: the number of attributes at the antecedent of rule i. $n(i_{con})$: the number of attributes at the consequent of rule i. ${\chi_i}^2$: chi-squared value of rule i. $\alpha_{i_{new}}$: additional constant defined as

$$\alpha_{i_{new}} = \begin{cases} \alpha_{new} & (i \text{ is new}) \\ 0 & (i \text{ has been extracted already}) \end{cases} \quad (6)$$

At each generation, individuals are replaced with new ones by selection and reproduction rules. Each individual is ranked by fitness evaluation value and selected by ranking. New individuals are generated by crossover and mutation. These operators are executed at a part of judgement nodes and a part of processing nodes of GNP genes, respectively. We demonstrate the rule concretely using the case of 120 individuals at each generation. The individuals are ranked by fitness values and the top 40 individuals are selected. They are reproduced three times and three genetic operators are executed to them as follows:
Crossover : crossover we used is the uniform crossover, and it is executed between two parents and generates two offspring. Each judgement node is selected as a crossover node with the probability of P_c. Two parents exchange the genes of the corresponding crossover nodes. 40 individuals are divided into 20 pairs of parents and replaced with new 40 individuals.
Mutation-1 : Mutation-1 operator affects one individual. The connection of each judgement node is changed randomly by mutation rate of P_{m1}. Top 40 individuals reproduce new 40 individuals by Mutation1.
Mutation-2 : Mutation-2 operator also affects one individual. The connection is changed to barter the judgement nodes. For example, in Fig.3, if '$B = 1$' node is bartered with '$D = 1$' node in position, then we examine $A \Rightarrow D$, $A \wedge D \Rightarrow C$, $A \wedge D \wedge C \Rightarrow B$ and so on. Mutation-2 is executed using the rate of P_{m2} at each judgement node. New 40 individuals are reproduced by Mutation2. Table 4 shows samples of P_c, P_{m1} and P_{m2}. All the connections of processing nodes are changed randomly in order to extract rules efficiently.

Table 4: Conditions of crossover and mutation

	GNP-M	GNP-L
Crossover Probability (P_c)	15/78	10/78
Mutation-1 Probability (P_{m1})	25/78	15/78
Mutation-2 Probability (P_{m2})	16/78	12/78

(Note: 78 corresponds to the number of Judgement nodes)

5 Simulation Results

We have performed experiments and estimated the performance of our algorithms. All the experiments were run on synthetic data. The synthetic database includes 26 attributes (A_j, $j = 1, 2, \ldots, 26$). The number of tuples are 200, $support(A_j = 1) = 0.7$ ($j = 1, 2, \ldots, 5$) and $support(A_j = 1) = 0.5$ ($j = 6, 7, \ldots, 26$). Evaluation is studied in the case of
free consequent (Simulation 1)
fixed consequent (Simulation 2)
in order to analyze the performance of rule extraction. In simulations, the population size is 120. The number of processing nodes is 10, and 26 different kind of judgement nodes ('$A_j = 1$', $j = 1, 2, \ldots, 26$) are used, each by three. We use (3), (4), (5) and (6) ($\alpha_{new} = 150$). Table 4 shows the two conditions of crossover and mutation. In addition, we consider the Random GNP model, which does not evolve but repeats random initialization every generation.

5.1 Simulation1

We have performed two experiments as follows:
1) $sup_{min} = 0.2$, $n(i_{ante}) \leq 5$, $n(i_{con}) \leq 5$

Table 5: Number of association rules of free consequent in the pool (Simulation 1)

		25^{th} generation			100^{th} generation			1000^{th} generation		
		GNP-M	GNP-L	Random	GNP-M	GNP-L	Random	GNP-M	GNP-L	Random
$sup_{min} = 0.2$	Max	748	741	616	905	903	808	922	922	922
	Ave	714.8	708.3	604.3	898.9	868.6	794.4	921.9	921.3	922.0
	Min	658	678	593	889	833	781	921	920	922
$sup_{min} = 0.1$	Max	97	131	57	1000	1305	181	4821	3696	1323
6 or more	Ave	55.3	68.8	31.8	300.5	918.5	136.1	2834.5	3029.2	1213.6
attributes	Min	20	36	20	148	549	111	1568	1121	1139

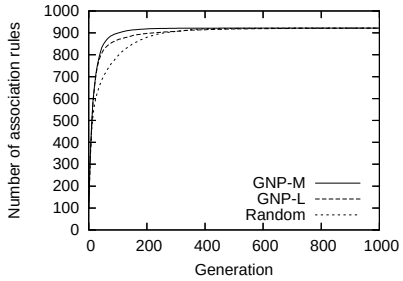

Figure 4: Number of association rules of free consequent ($sup_{min} = 0.2$)

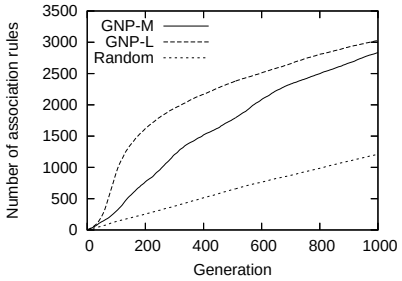

Figure 5: Number of association rules of free consequent ($sup_{min} = 0.1$, 6 or more attributes)

2) $sup_{min} = 0.1$, $n(i_{ante}) + n(i_{con}) \geq 6$, $n(i_{ante}) \leq 5$, $n(i_{con}) \leq 5$

The number of changing the connections of processing nodes is 5. Table 5, Fig.4 and Fig.5 show the number of important association rules in the pool. The system can extract the important association rules in the database effectively. Each figure shows the mean value over ten simulations. Fig.6 and Fig.7 show fitness curve. Fig.8 and Fig.9 are the results of the number of association rules at each generation. These show the different important association rules in 120 individuals at each generation. GNP-M suits the extraction of the rules including 3-5 attributes, while GNP-L will be convenient for 5-7 attributes rules. The results show that our method works effectively by its fitness curve and the number of extracted rules at each generation. It is found that the proposed evolutionary method is effective in association rule mining. In addition, it is also found that we can set a condition to extract rules, for instance, the number of attributes in the rules.

5.2 Simulation2

We have performed experiments with one specific consequent attribute (A_{26}) supposing $sup_{min} = 0.05$ and $n(i_{ante}) \leq 8$. (such as $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \wedge A_5 \Rightarrow A_{26}$ will be extracted.) As we suppose that the $support(A_{26} = 1) = 0.5$, the method is fairly simplified. Each judgement node examines the '$A_{26} = 1$'. For example, in Fig.3, the number of tuples 'b' indicates $support((A = 1) \wedge (B = 1))$, and GNP calculates $support((A = 1) \wedge (B = 1) \wedge (A_{26} = 1))$ at the same time, because $support(A_{26} = 1)$ is known. Then we can obtain the measurements of the rule $A \wedge B \Rightarrow A_{26}$. Experiments is performed by $P_c = 15/75$, $P_{m1} = 25/75$, $P_{m2} = 16/75$. (judgement node '$A_{26} = 1$' is not used.) Fig.10 shows the number of important association rules. Especially, Fig.11 shows the number of rules satisfying $n(i_{ante}) \geq 7$. The results also show that our method works effectively. It is also found that the system can extract the interesting rules easily, which are made up of 7 or more antecedent attributes.

6 Conclusions

In this paper, a new method of association rule mining using Genetic Network Programming (GNP) has been proposed. The proposed system can evolve itself by an evolutionary method and measures the significance of associations via the chi-squared value. An efficient algorithm for identifying association rules of importance was designed. We have performed experiments and estimated the performance of our algorithms. The results showed that our method extracts the important association rules in the database effectively. In addition, it is found that we can set a condition to extract rules, for instance, the number of attributes in the rules. In a future, we plan to extend the proposed method to the one applicable to large databases.

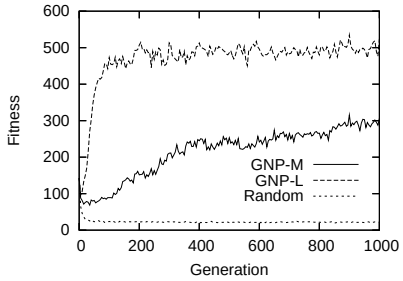

Figure 6: Fitness curves of free consequent ($sup_{min} = 0.2$)

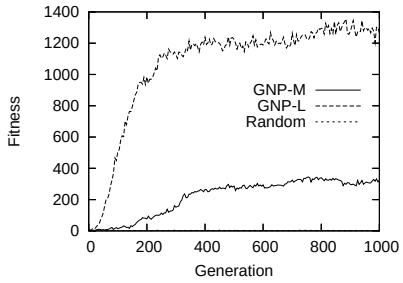

Figure 7: Fitness curves of free consequent ($sup_{min} = 0.1$, 6 or more attributes)

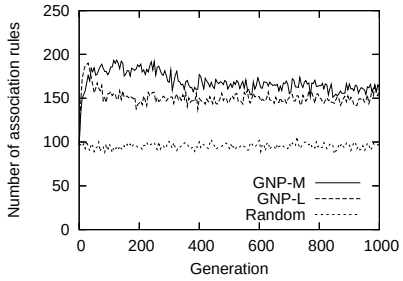

Figure 8: Number of association rules of free consequent ($sup_{min} = 0.2$)

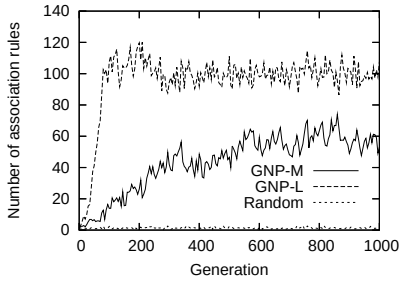

Figure 9: Number of association rules of free consequent ($sup_{min} = 0.1$, 6 or more attributes)

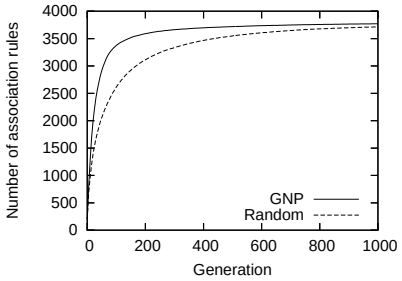

Figure 10: Number of association rules of fixed consequent ($sup_{min} = 0.05$)

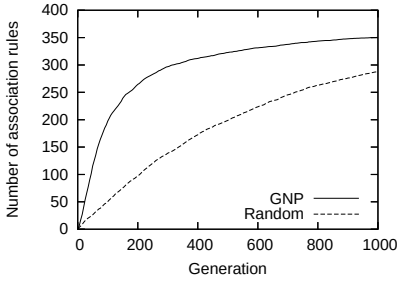

Figure 11: Number of association rules of fixed consequent ($sup_{min} = 0.05$, 7 or more attributes)

References

[1] C. Zhang , S. Zhang , Association Rule Mining: models and algorithms , Springer (2002)

[2] R. Agrawal, R. Srikant : "Fast Algorithms for Mining Association Rules ", *Proc. of the 20th VLDB Conf.* Santiago , Chile, pp.487-499 (1994)

[3] S. Brin, R. Motwani, and C. Silverstein. "Beyond market baskets : generalizing association rules to correlations " *In Proc. of ACM SIGMOD*, pp.265-276 (1997)

[4] H. Katagiri, K. Hirasawa, and J. Hu : "Genetic network programming - application to intelligent agents -", *In Proc. of IEEE International Conf. on Syst. , Man and Cybernetics*, pp.3829-3834 (2000)

[5] H. Katagiri, K. Hirasawa, J. Hu and J. Murata: "Network Structure Oriented Evolutionary Model - Genetic Network Programming", *in Proc. of Genetic and Evolutionary Computation Conference*, pp.219-226 (2001)

[6] K. Hirasawa, M. Okubo, H. Katagiri, J. Hu and J. Murata: "Comparison between Genetic Network Programming (GNP) and Genetic Programming (GP)", *in Proc. of Congress of Evolutionary Computation*, pp.1276-1282 (2001)