Lymphoma Cancer Classification Using Genetic Programming with SNR Features

Jin-Hyuk Hong and Sung-Bae Cho

Dept. of Computer Science, Yonsei University,
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
hjinh@candy.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Lymphoma cancer classification with DNA microarray data is one of important problems in bioinformatics. Many machine learning techniques have been applied to the problem and produced valuable results. However the medical field requires not only a high-accuracy classifier, but also the in-depth analysis and understanding of classification rules obtained. Since gene expression data have thousands of features, it is nearly impossible to represent and understand their complex relationships directly. In this paper, we adopt the SNR (Signal-to-Noise Ratio) feature selection to reduce the dimensionality of the data, and then use genetic programming to generate cancer classification rules with the features. In the experimental results on Lymphoma cancer dataset, the proposed method yielded 96.6% test accuracy in average, and an excellent arithmetic classification rule set that classifies all the samples correctly is discovered by the proposed method.

1 Introduction

Accurate decision and diagnosis of the cancer are very important in the field of medicine while they are very difficult [1,2]. Exact classification of cancers makes it possible to treat a patient with proper treatments and helpful medicines so as to save the patient's life. Over several centuries, various cancer classification techniques are developed, but most of them are based on the clinical analysis of morphological symptoms for the cancer. With these methods, even a medical expert causes many errors and misunderstandings, because in many cases different cancers show some similar symptoms. In order to overcome these restrictions, classification techniques using human's gene information have been actively investigated, and many good results have been reported recently [1,2,3]

Gene information, usually called gene expression data, is collected by the DNA microarray technique with keen interests. The gene expression data include lots of gene information on living things [2]. Usually, the gene expression data provide useful information for the classification of different kinds of cancers. Since the original format of the data is an array of simple numbers, it is not easy to analyze them directly and to discover useful classification rules of the cancer. Several methods for it have been studied for several years in artificial intelligence [2,3]. Table 1 shows related works on the classification of lymphoma cancer using DNA microarray data.

M. Keijzer et al. (Eds.): EuroGP 2004, LNCS 3003, pp. 78–88, 2004.

Table 1. Related works

Author	Data	Method		Accuracy (%)
		Feature selection	Classifier	
Li et al.	Lymphoma	Genetic algorithm	Knn	84.6
Dudoit et al.		The ratio of between-groups to within-groups sum of squares	Nearest neighbor	95.0
			Diagonal linear discriminant analysis	95.0
Nguyen et al.		PCA	Logistic discriminant	98.1
			Boost CART	97.6

It is not easy to obtain a good classification performance with gene expression data, because the data consist of a few samples with a large number of variables. Nevertheless diverse technologies of artificial intelligence have been applied to classify the cancer and shown a superior performance of the classification. However, many conventional approaches such as the neural network and SVMs are not easy to be directly interpreted. In medical area discovered rules should be understandable for people to get a confidence [4]. In this paper, we propose a classification rule generation method which is composed of the SNR feature selection and genetic programming so as to obtain precise and comprehensible classification rules, which also produces an outstanding performance from high dimensional gene expression data by designing the rule with arithmetic operations.

2 Backgrounds

2.1 DNA Microarray

An organism basically has thousands of genes, RNA and protein. Traditional molecular biology has only considered a single gene, so the obtained information is very limited to be applied various problems. DNA microarray has been developed recently, and it successfully deals with the problem. It acquires gene information in terms of microscopic units, and the revelation phase of a total chromosome on a chip is observed by this technique. That is, DNA microarray technique makes it possible to analyze and observe for a complex organism in detail [1,2,3].

DNA microarray fixes cDNA of high density on a solid substrate which is not permeated with a solution, while it attaches thousands of DNA and protein at regular intervals on the solid substrate and combines with the target materials. The phase of the combination can be observed on the chip. Each cell on the array is synthesized with two gene materials collected by different environments and different fluorescent dyes mixed (green-fluorescent dye Cy3 and red-fluorescent dye Cy5 in equal quantities). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged by a scanner that makes the fluorescence measurement for each dye. The overall procedure of DNA microarray technology is as shown in Fig.1 and the log ratio between the two intensities of each dye is used as the gene expression as follows.

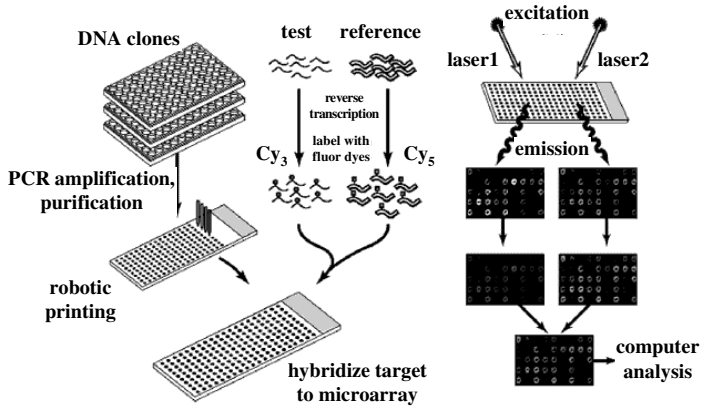


Fig. 1. Overview of DNA microarray technology

$$gene\_expression = \log_2 \frac{\text{Int}(Cy5)}{\text{Int}(Cy3)}$$

where Int(Cy5) and Int(Cy3) are the intensities of red and green colors. Since at least hundreds of genes are put on the DNA microarray, we can investigate the genome-wide information in short time.

2.2 Genetic Programming

Genetic programming is devised to design a program which solves a problem automatically without a user's explicit programming. It regards a program as a structure composed of functions and variables. The program usually has a tree structure to represent the individual's information [13].

Genetic programming is one of evolutionary computation techniques like the genetic algorithm. Basic operations and characteristics are similar to those of the genetic algorithm, but they are different in terms of the representation. The solution space of genetic programming is very wide reaching to problems which can be solved by a program with functions and variables [10,11,14]. There are various functions for genetic programming such as arithmetic operations, logical operations, and user-defined operations. Recently, it has been applied to many problems such as optimization, the evolution of assembly language program, evolvable hardware, the generation of a virtual character's behaviors, etc [13].

3 Classification Rule Discovery

In this paper, we propose a rule discovery method as shown in Fig. 2. First, the SNR feature selection reduces the dimensionality. And then, genetic programming finds out good classification rules with the SNR features.

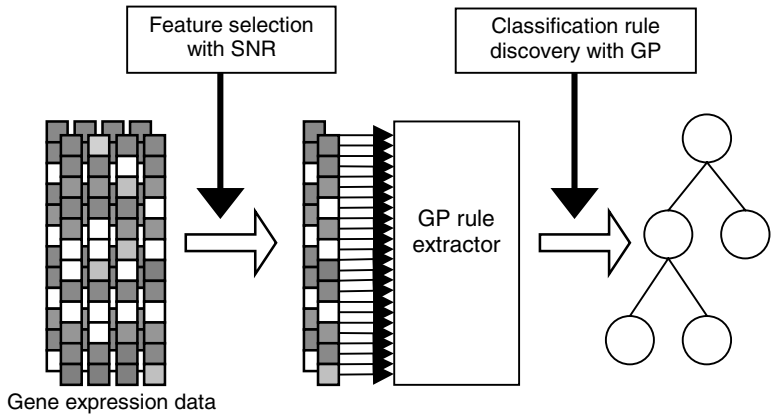


Fig. 2. The proposed method to classify DNA microarray profiles

3.1 Signal-to-Noise Ratio Feature Selection

Since not all the genes are associated with a specific disease, the feature selection often called gene selection is necessary to extract informative genes for the classification of the disease [3,15,16]. Moreover, feature selection accelerates the speed of learning a classifier and removes noises in the data.

There are two major feature selection approaches: filter and wrapper approaches. The former selects informative features (genes) regardless of classifiers. It independently measures the importance of features, and selects some for the classification. On the other hand, the latter selects features together with classifiers. It is simultaneously done by the training of a classifier to produce the optimal combination of features and a classifier. Since the filter approach is simple and fast enough to obtain high performance, we evaluated various filter-based feature selection methods [15]. Finally signal-to-noise ratio ranking method is adopted to select useful features. After measuring the signal to noise ratio of genes, 30 genes are selected based on their ranks.

$$SN(g_i, C) = \frac{\mu_{c1}(g_i) - \mu_{c0}(g_i)}{\sigma_{c1}(g_i) + \sigma_{c0}(g_i)}$$

$$\mu_1(g) : \textit{the average of genes in class } C$$

$$\mu_2(g) : \textit{the average of genes not in class } C$$

$$\sigma_1(g) : \textit{standard deviation of genes in class } C$$

$$\sigma_2(g) : \textit{standard deviation of genes not in class } C$$

Signal-to-noise ratio measures how the signal from the defect compares to other background noise. In bioinformatics the signal represents useful information conveyed by genes, and noise to anything else on the genes. Hence a low ratio implies that the gene is not worth for the class *C* while a high ratio means that the gene is rather related with the class *C*.

Table 2. Arithmetic operators used in this paper

Arithmetic operator	Function	Description
+	Addition	Positive effect on class 1(Negative effect on class 2)
–	Subtraction	Negative effect on class 1(Positive effect on class 2)
×	Multiplication	Multiplicative correlation
/	Division	Divisive correlation

3.2 Classification Rule Extraction

Conventional rule discovery using genetic programming has usually adopted first-order logic [17] or IF-THEN structure as the rule, while logic operations AND, OR, Not and comparative operations (<, >, =) are frequently used as follows [4,12].

$$Rule1: \text{ IF}((A1 < 0.6) \text{ OR } (A3 > 0.3)) \text{ THEN } class1$$
$$Rule2: \text{ IF}((A2 = 0.7) \text{ AND } (A1 > 0.7)) \text{ THEN } class2$$

Although these rules are easy to be interpreted, it has a limitation to represent more complex relationships among variables to get a high performance [12]. Mathematical operations have been also tried to construct a rule, but they are difficult in the analysis. Moreover in some applications it is already known that they obtain lower accuracy than arithmetic operations.

In this paper, arithmetic operations are used to construct a more sophisticated rule leading to high accuracy. A rule is designed as a tree with 30 SNR features and basic arithmetic operations (+, -, ×, /). Although numerical value can be also considered as a terminal, it is not used in this experiment. For the easy analysis of rules obtained, the meanings of arithmetic operations for genes are defined in Table 2.

The classification rule is constructed as follows. As shown in Fig. 3, the value of the function eval() represents which class a sample belongs to. Positive value indicates that the sample belongs to class 1, while negative value signifies that the sample is classified into class 2.

$$\text{IF } eval(Individual_i) \geq 0 \text{ THEN } class1 \text{ ELSE } class2$$

We have experimented with three kinds of rule representations. Not all arithmetic operators are used as shown in Table 3, but the 2nd and the 3rd without × and / operators are used to keep the simplicity of the rule. Weights show that which gene is more effective for the classification while the values are from 0 to 1.0. Fig. 4 briefly shows the three rule representations.

Table 3. Rule representations to be tested

No	+	-	×	/	Weighting	Complexity
1	Use	Use	Use	Use	Not-use	High
2	Use	Use	Not-use	Not-use	Use	Middle
3	Use	Use	Not-use	Not-use	Not-use	Low

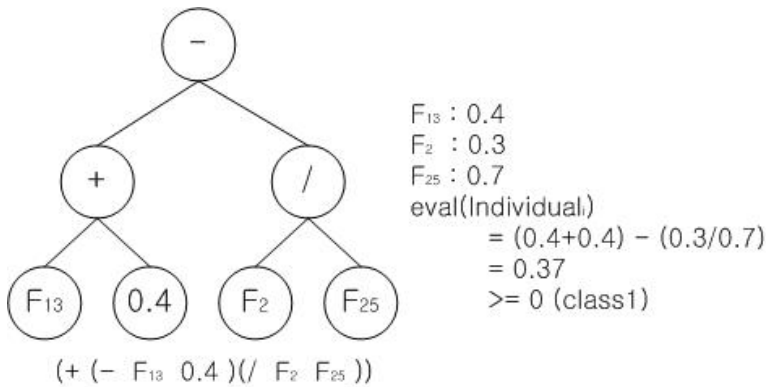


Fig. 3. Representation of the proposed method and classification rule

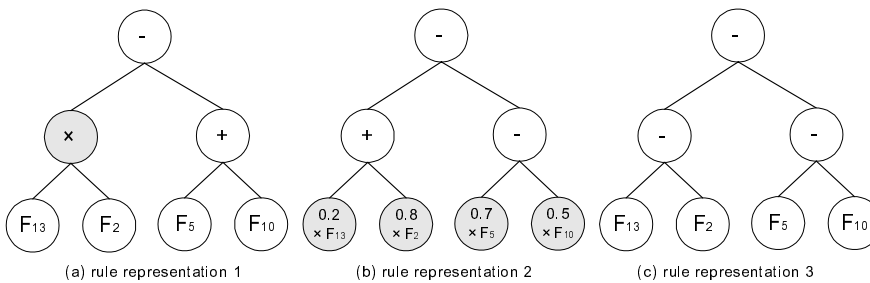


Fig. 4. 3 different rule representations

The performance for the training data is used as the fitness of a rule. The simplicity measure is added on the fitness function to get comprehensible-sized classification rules as follows. It is generally known that a simpler classifier is more general than complicated one with the same accuracy for the training data.

$$fitness\ \ of\ \ individual_i = \frac{\text{number of correct samples}}{\text{number of total train data}} \times w_1 + simplicity \times w_2$$

$$\text{where}\ \ simplicity = \frac{\text{number of nodes}}{\text{number of maximum nodes}},$$

$$w_1 = \text{weight for training rate, and}\ w_2 = \text{weight for simplicity}$$

Table 4. Experimental environments

Parameter	Value (final)	Parameter	Value (final)
Population size	100	Mutation rate	0.1~0.3 (0.2)
Maximum generation	50,000	Permutation rate	0.1
Selection rate	0.6~0.8 (0.8)	Maximum depth of a tree	3
Crossover rate	0.6~0.8 (0.8)	Elitism	yes

4 Experiments

4.1 Experimental Environment

The proposed method is verified with Lymphoma cancer dataset, which is well known microarray dataset [18]. This dataset (http://llmpp.nih.gov/lymphoma/) is one of popular DNA microarray datasets used in bioinformatics for the benchmark. It consists of 47 samples: 24 samples of GC B-like and 23 samples of activated B-like. Each sample has 4,026 gene expression levels. All features are normalized from 0 to 1.

Since the gene expression data consist of few samples with many features, the proposed method is evaluated by leave-one-out cross-validation. Total 47 experiments are conducted, where each sample is set as the test data and the others are set as the train data. All experiments are repeated 10 times and the average of them is used as the final result.

The parameters for genetic programming are set as shown in Table 4. We use roulette wheel selection with elite preserving strategy, and set the weights w_1 and w_2 of the fitness evaluation function as 0.9 and 0.1, respectively.

4.2 Results Analysis

Fig. 5 shows the accuracy for the test data in terms of the rule representations. We can get 96.6% test accuracy in average with the third rule representation although this is the simplest among the three rule representations.

Fig. 6 shows the classification rules which are the most frequently occurred in the experiments, while they classify all the samples correctly with a few genes. The detailed descriptions of the genes are shown in Table 5~7. The functions of some genes are not known yet, and this gives interest to medical experts to study the functions of those genes. Although the rules are obtained by the cross-validation, we focus on the easy interpretability and the information included in the rules.

The rule shown in Fig. 6(a) is analyzed based on the meaning of the arithmetic operations as described in Table 1. F4 affects a sample to be included into class 2 while negatively into the class 1. F20 and F25 are combined by a multiplicative correlation, so as to push samples to be classified into class1. We can interpret it as follows so to obtain some information from the rule:

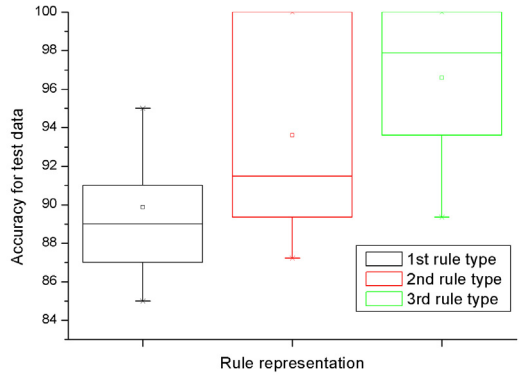
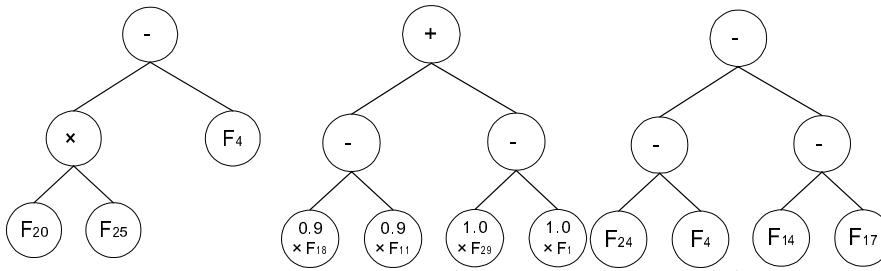


Fig. 5. The accuracy for test data



(a) 1st rule (b) 2nd rule (c) 3rd rule

Fig. 6. The rules for perfect classification with each rule representation

· F4, F20, and F25 are related with lymphoma cancer
· The value of F4 is negatively related with the cancer
· F20 and F25 are positively related with the cancer

We have conducted an additional experiment to compare the proposed method with a neural network, one of promising machine learning techniques. 3-layered multi-layer perceptron is used with 2~10 hidden nodes, 2 output nodes, learning rate of 0.01~0.1 and momentum of 0.7~0.9. The maximum iteration for learning is fixed to 5000. Three features are used as the input of the neural network. The training accuracy is 98%, while the test accuracy is 97.8%. Even with intensive efforts, we could not get 100% accuracy with the neural network. The neural network has been also learned with 30 features, but the result is worse than the first case. It just obtained 95.7% training accuracy and 95.7% test accuracy. This proves that genetic programming also selected useful features among the 30 features. The additional experiment shows the competitive performance of the proposed method in the classification of the dataset.

Fig. 6(b) and Fig. 6(c) are rules for the 2nd and the 3rd rule representations. Based on the analysis method, each classification rule includes the following information.

Table 5. The detailed description of genes used in the rule shown in Fig. 6(a)

Feature #	Gene #	Description
F20	75	Unknown UG Hs.169081 ets variant gene 6 (TEL oncogene); Clone=1355435, 14671
F25	2467	*core binding factor alpha1b subunit=CBF alpha1=PEBP2aA1 transcription factor =AML1 Proto-oncogene=translocated in acute myeloid leukemia; Clone=263251, 17823
F4	1277	Unknown UG Hs.136345 ESTs; Clone=746300, 19274

Table 6. The detailed description of the genes used in the rule shown in Fig. 6(b)

Feature #	Gene #	Description
F18	1636	CXCR5=BLR1=B-cell homing chemokine receptor=L1; Clone=31, 4297
F11	1246	*FAK=focal adhesion kinase; Clone=795352, 17333
F29	86	*BCL-2; Clone=342181, 17646
F1	1268	*CD10=CALLA=Neprilysin=enkepalinase; Clone=200814, 15864

Table 7. The detailed description of genes used in the rule shown in Fig. 6(c)

Feature #	Gene #	Description
F24	684	Unknown; Clone=1352715, 14377
F4	1279	*Unknown; Clone=825199, 19288
F14	1914	Lymphotoxin-Beta=Tumor necrosis factor C; Clone=1320296, 13297
F17	680	*Unknown; Clone=1372162, 19541

The classification rule in Fig. 6(b) can be interpreted as follows:
· F18, F11, F29, and F1 are related with the lymphoma cancer
· F18 and F29 affect positively on the GC B-like lymphoma cancer
· F11 and F1 are negatively related with the GC B-like lymphoma cancer
· Each weight signifies the importance on the cancer classification
The classification rule in Fig. 6(c) can be interpreted as follows:
· F24, F4, F14, and F17 are related with the lymphoma cancer
· F24 and F17 affect positively on the GC B-like lymphoma cancer
· F4 and F14 are negatively related with the GC B-like lymphoma cancer

F29 used in the 2nd rule is the *BCL-2 gene, which turned out that it is related with the lymphoma cancer [19]. F14 described in Table 6 is known that it relates with the lymphoma cancer. These imply that the rules discovered by the proposed method are understandable, and there is a possibility that the other features are related with the lymphoma cancer. These rules also need a demonstration by medical experts, but there is a good chance of discovering useful information from them.

5 Concluding Remarks

In this paper, we have proposed an effective rule generation method, which uses genetic programming with SNR features. Since gene expression data have huge-scale feature data with a few samples, it is difficult to generate valuable classification rules from the data directly. The SNR feature selection method used in this paper remarkably reduces the number of features, while genetic programming generates useful rules with those features selected. Moreover we have proposed the analysis method for the arithmetic rule representation. It is very simple but helpful for the interpretation of the rules extracted. The experimental results show that the performance of the proposed method is effective to extract classification rules with 96.6% test accuracy, and also good classification rules have been easily interpreted and provided useful information for the classification.

As the future work, we will verify the obtained results with medical experts and try to combine logical and arithmetic structures in genetic programming for better classification. Each structure has its advantage, and the combination might help to improve the performance and interpretability.

Acknowledgements. This work was supported by Biometrics Engineering Research Center, and a grant of Korea Health 21 R&D project, Ministry of Health & Welfare, Republic of Korea.

References

1. A. Ben-Dor, et al., "Tissue classification with gene expression profiles," *J. of Computational Biology*, vol. 7, pp. 559-584, 2000.
2. A. Brazma and J. Vilo, "Gene expression data analysis," *Federation of European Biochemical Societies Letters*, vol. 480, pp. 17-24, 2000.
3. C. Park and S.-B. Cho, "Genetic search for optimal ensemble of feature-classifier pairs in DNA gene expression profiles," *Int. Joint Conf. on Neural Networks*, pp. 1702-1707, 2003.
4. K. Tan, et al., "Evolutionary computing for knowledge discovery in medical diagnosis," *Artificial Intelligence in Medicine*, vol. 27, no. 2, pp. 129-154, 2003.
5. J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
6. D. Goldberg, *Genetic Algorithms in Search, Optimaization, and Machine Learning*, Addison-Wesley, 1989.
7. K. DeJong, et al., "Using genetic algorithms for concept learning," *Machine Learning*, vol. 13, pp. 161-188, 1993.
8. A. Freitas, "A survey of evolutionary algorithms for data mining and knowledge discovery," *Advances in Evolutionary Computation*, pp. 819-845, 2002.

9. C. Hsu and C. Knoblock, "Discovering robust knowledge from databases that change," *Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 69-95, 1998.
10. C. Zhou, et al., "Discovery of classification rules by using gene expression programming," *Proc. of the 2002 Int. Conf. on Artificial Intelligence*, pp. 1355-1361, 2002.
11. C. Bojarczuk, et al., "Discovering comprehensible classification rules using genetic programming: A case study in a medical domain," *Proc. of the Genetic and Evolutionary Computation Conf.*, pp. 953-958, 1999.
12. I. Falco, et al., "Discovering interesting classification rules with genetic programming, " *Applied Soft Computing*, vol. 1, no. 4, pp. 257-269, 2002.
13. J. Koza, "Genetic programming," *Encyclopedia of Computer Science and Technology*, vol. 39, pp. 29-43, 1998.
14. J. Kishore, et al., "Application of genetic programming for multicategory pattern classification," *IEEE Trans. on Evolutionary Computation*, vol. 4, no. 3, pp. 242-258, 2000.
15. H.-H. Won and S.-B. Cho, "Neural network ensemble with negatively correlated features for cancer classification, " *Lecture Notes in Computer Science*, vol. 2714, pp. 1143-1150, 2003.
16. J. Bins and B. Draper, "Feature selection from huge feature sets," *Proc. Int. Conf. Computer Vision 2*, pp. 159-165, 2001.
17. S. Augier, et al., "Learning first order logic rules with a genetic algorithm," *Proc. of the First Int. Conf. on Knowledge Discovery & Data Mining*, pp. 21-26, 1995.
18. A. Alizadeh, et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, " *Nature*, vol. 403, pp. 503-511, 2000.
19. O. Monni, et al. "BCL2 overexpression in diffuse large B-cell lymphoma," *Leuk Lymphoma*, vol. 34, no 1-2, pp. 45-52, 1999.