양자 유전알고리즘을 이용한 특징 선택 및 성능 분석

(Feature Selection and Performance Analysis using Quantum-inspired Genetic Algorithm)

허기수[*], 정현태[*], 박아론[**], 백성준[**]

(G.S. Heo, H.T. Jeong, A. Park, S.J. Baek)

요약

특징 선택은 패턴 인식의 성능을 향상시키기 위해 부분집합을 구성하는 중요한 문제다. 특징 선택에는 순차 탐색 알고리즘으로부터 확률 기반의 유전 알고리즘까지 다양한 접근 방법이 적용 되었다. 본 연구에서는 특징 선택을 위해 양자 비트, 상태의 중첩 등 양자 컴퓨터 개념을 기반으로 하는 양자 기반 유전 알고리즘(QGA: Quantum-inspired Genetic Algorithm)을 적용하였다. QGA 성능은 전통적인 유전 알고리즘(CGA: Conventional Genetic Algorithm)을 적용한 특징 선택 방법과 분류율 및 평균 특징 개수의 비교를 통해 이루어졌으며, UCI 데이터를 이용한 실험 결과 QGA를 적용한 특징 선택 방법이 CGA를 적용한 경우에 비해 전반적으로 좋은 성능을 보임을 확인 할 수 있었다.

Abstract

Feature selection is the important technique of selecting a subset of relevant features for building robust pattern recognition systems. Various methods have been studied for feature selection from sequential search algorithms to stochastic algorithms. In this work, we adopted a Quantum-inspired Genetic Algorithm (QGA) which is based on the concept and principles of quantum computing such as Q-bits and superposition of state for feature selection. The performance of QGA is compared to that of the Conventional Genetic Algorithm (CGA) with respect to the classification rates and the number of selected features. The experimental result using UCI data sets shows that QGA is superior to CGA.

■ keyworld : | Feature Selection | Conventional Genetic Algorithm | Quantum-inspired Genetic Algorithm |

Ⅰ. Introduction

Feature selection is the problem of selecting a subset of features from a total of features based on some optimization criterion. The primary purpose of feature selection is to design a more compact classifier without any performance degradation. The reduced number of features helps to build less computational demanding applications in the mobile environment.

Since the feature selection is a typical combinatorial optimization problem, it is essential to develop an algorithm to efficiently search in a wide range of search space [1]. Accordingly, many feature selection algorithms have been studied such as enumeration algorithms, sequential search algorithms, and genetic algorithms (GA).

GA which is a stochastic algorithm based on the principles of natural biological evolution have shown capabilities in solving optimization problem in various science and engineering areas [2]. One of the important issues in GA is a balance between exploration and exploitation. There are many arguments to control this balance, but it is very difficult to find appropriate arguments. Recently, QGA (Quantum-inspired Genetic Algorithm) was proposed in [3,4], which could handle the balance between exploration and exploitation more easily when compared with CGA (Conventional Genetic

[*] 정회원, 한국전자통신연구원

[**]정회원, 전남대학교 전자컴퓨터공학과, tozero@chonnam.ac.kr

접수번호 : 2012-02-06-0002

접수일자 : 2012년 02월 06일

심사완료일 : 2012년 03월 07일

교신저자 : 허기수, e-mail : heogisu@etri.re.kr

Algorithm).

In this paper, we introduced QGA to solve the feature selection problem and compared the results with CGA. The experiments were carried out with UCI data sets and the performance of QGA and CGA were compared in terms of classification rates and the number of selected features.

Ⅱ. Genetic Algorithms for feature selection

1. Conventional Genetic Algorithm

The structure of CGA can be described as follows.

```
Conventional Genetic Algorithm

initialize population $P$;
evaluate $P$;
repeat   {
    select two parents $p_1$ and $p_2$ from $P$;
    offspring =   crossover($p_1$,$p_2$);
    mutation(offspring);
    replace offspring in $P$;
} until (stopping condition);
return the best chromosome obtained so far;
```

Figure 1. Procedure of CGA

For the feature selection, a string which is a chromosome with D binary digits is used. A binary digit describes the presence of the corresponding feature. Each chromosome in the population $p(=20)$ is evaluated by Multi-Layer Perceptron (MLP) classifier to give a measure of its fitness. The presence of input node of MLP classifier depends on the state of corresponding bits of the chromosome.

In 'select' step, we select two parent chromosomes by adopting the rank-based roulette-wheel selection scheme. Then crossover operation generates a new chromosome (offspring) out of the two parents, and the mutation operation slightly perturbs the offspring. We used the standard 2-point crossover and mutation operations with mutation rate $p_m(=0.01)$.

In 'replace' step, if a mutated chromosome is superior to both parents, it replaces the similar parent; if it is in between the two parents, it replaces the inferior parent. Otherwise, the most inferior chromosome in the population is replaced. Finally, GA stops when the number of generations reaches the given maximum generation $t(=100)$.

2. Quantum-inspired Genetic Algorithm

Quantum-inspired Genetic Algorithm (QGA) is a probabilistic algorithm similar to a genetic algorithm. It is based on the concept and principles of quantum computing such as Q-bits and superposition of state, and exploits Q-bit chromosome as a presentation. The smallest unit of information stored in two-state quantum computer is called a quantum bit or Q-bit, which may be in the 1 state or in the 0 state, or in any superposition of the two at the same time [5].

The state of a quantum bit can be represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad (1)$$

where α and β are numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ and $|\beta|^2$ give the probabilities that the Q-bit will be found in the 0 sate and 1 state, respectively. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1 \qquad (2)$$

QGA uses a novel representation that is based on the concept of Q-bits. One Q-bit is defined with a pair of numbers (α, β). A Q-bit chromosome as a string of m Q-bits is defined as

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right] . \qquad (3)$$

Q-bit representation has the advantage that it is able to represent a superposition of states. This idea of superposition makes it possible to represent an exponential set of state with a small number of Q-bits. The structure of QGA is described in the following.

```
Quantum-inspired   Genetic Algorithm

$t \leftarrow 0$;
initialize $Q(t)$;
make $P(t)$ by observing $Q(t)$ states;
evaluate $P(t)$;
store all solutions in $P(t)$ into $B(t)$ ;
repeat {
    $t \leftarrow t$ +1;
    make $P(t)$ by observing $Q(t-1)$ states;
    evaluate $P(t)$;
    update $Q(t)$ using quantum gates $U(\triangle\theta)$;
    store the best solutions in $B(t)$ and
    $P(t)$ into $B(t)$;
    store the best solution $\hat{\theta}$ among $B(t)$;
} until (stopping condition);
return the best solution $\hat{\theta}$;
```

Figure 2. Procedure of QGA

QGA maintains a population of Q-bit chromosomes, $Q(t)=\{q_1^t, q_2^t, \cdots, q_n^t\}$ at generation $t(=100)$, where $n(=20)$ is the size of population, and q_j^t is a Q-bit chromosome defined as

$$q_j^t = \left[\begin{array}{c|c|c|c} \alpha_{j1}^t & \alpha_{j2}^t & \cdots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \cdots & \beta_{jm}^t \end{array} \right] \tag{4}$$

where m is the string length of the Q-bit chromosome.

In the step of 'initialize $Q(t)$', α_i^t and β_i^t of all q_j^t are initialized with $1/\sqrt{2}$. It means that the first Q-bit chromosome represents the linear superposition of all possible states with the same probability.

The next step makes a set of binary solutions $P(t)=\{p_1^t, p_2^t, \cdots, p_n^t\}$ by observing $Q(t)$. One binary solution p_j^t is a binary string of length m and is formed by selecting either 0 or 1 for each bit using the probability, $|\alpha_i^t|^2$ or, $|\beta_i^t|^2$ respectively. For every bit in the binary string, we generate a random number r from the range [0,1]; if $r > |\alpha_i|^2$, we set the bit of the binary string.

Each binary solution p_j^t is evaluated by MLP classifier to give a measure of its fitness like in CGA. The best solution among $P(t)$ and $B(t-1)$ is then selected and stored into $B(t)=\{b_1^t, b_2^t, \cdots, b_n^t\}$.

In the step of 'update $Q(t)$,' Q-bit chromosomes in $Q(t)$ are updated by applying some appropriate quantum gates $U(\theta)$, which is formed by using the binary solutions $P(t)$ and $B(t)$. This Q-gate has the same role as the crossover and mutation operators of CGA. In this work, a Q-bit chromosome q_j^t is updated by using the following rotation gate $U(\theta)$.

$$\mathrm{U}(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \tag{5}$$

Where θ_i is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign. θ_i should be designed in compliance with the application problem. In this problem, θ_i is given as $s(\alpha_i \beta_i)\triangle\theta_i$. The parameters used are shown in Table 1. For example, if $f(p) \geq f(b)$ is satisfied p_j^t and b_j^t are 1 and 0, respectively, we can set the value of $\triangle\theta_i$ as 0.025π and $s(\alpha_i \beta_i)$ as +1, -1 or 0 according to the condition of $\alpha_i \beta_i$. The sign $s(\alpha_i \beta_i)$ determines the direction of convergence and $\triangle\theta_i$ determines the speed of convergence.

```
Q-bit chromosome update:
for ( $i$ = 1 to $m$ )
    determine $\theta_i$ with the lookup table
    obtain $(\alpha_i', \beta_i')$as:
    $[\alpha_i', \beta_i']^T = U(\theta_i)[\alpha_i, \beta_i]^T$
```

Table 1. Lookup table of θ_i

p_i	b_i	$f(p) \ge f(b)$	$\Delta\theta_i$	$s(\alpha_i\beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	false	0	0	0	0	0
0	0	true	0	0	0	0	0
0	1	false	0	0	0	0	0
0	1	true	0.05π	-1	+1	±1	0
1	0	false	0.01π	-1	+1	±1	0
1	0	true	0.025π	+1	-1	0	±1
1	1	false	0.005π	+1	-1	0	±1
1	1	true	0.025π	+1	-1	0	±1

Ⅲ. Experimental Environment and Result Analysis

1. Environment

We used 5 data sets from UCI Machine Learning Repository [6], which offers various real world classification problems. Table 2 shows the specification of the data sets. These data sets possess diverse characteristics in terms of the number of samples, features, and classes. All the data was normalized by a linear function before usage.

Table 2. Data sets used for experiment and MLP structure

Data set	Number of samples	Number of features	Number of classes	MLP structure
Pima-Indians	768	8	2	8-10-2
Segmentation	2310	19	7	19-10-7
WDBC	569	30	2	30-10-2
Waveform-Noise	5000	40	3	40-10-3
Sonar	208	60	2	60-10-2

In the experiment, we used three layers MLP classifier. In the training, the well known error back-propagation algorithm is used. The number of input nodes is set equal to the number of features, and the number of output nodes is set equal to the number of classes.

We used a five-fold cross validation schema for the performance evaluation and stop the training when RMSE, which is evaluated by validation set, undergoes five consecutive increases.

2. Result Analysis

The experimental results are summarized in Table 3. The table shows the Best, Worst, and Average performance and the number of selected features with respect to the data sets and the feature selection methods.

The usefulness of feature selection can be easily verified with the experimental result. The performance with respect to the classification rates and the number of features in the table shows that the feature selection by CGA or QGA is superior to non-feature selection without regarding to the data sets.

Table 3. Comparison of the Non-feature selection and feature selection by CGA and QGA

			NFS	FS	
				CGA	QGA
Pima-Indians	Performance	B.	83.66	84.31	84.97
		W.	71.90	70.59	70.59
		A.	76.47	77.65	77.91
	Number of features		8	4.6	4.6
Segmentation	Performance	B.	99.57	100	100
		W.	98.70	98.27	99.13
		A	99.31	99.48	99.65
	Number of features		19	9.4	8
WDBC	Performance	B.	98.23	100	100
		W.	92.04	96.46	96.46
		A.	96.64	98.58	98.78
	Number of features		30	11.6	12.4
Waveform -Noise	Performance	B.	61.10	84.80	83.20
		W.	58.00	60.40	78.00
		A.	60.00	78.54	80.52
	Number of features		40	21.4	19.2
Sonar	Performance	B.	82.93	95.12	95.12
		W.	70.73	63.41	63.41
		A.	76.10	80.49	84.39
	Number of features		60	31.6	34.0

(NFS: Non-Feature Selection, FS: Feature Selection, B: Best, W: Worst, A: Average)

When we compare the performance between CGA and QGA, some tendencies are found. For the case of Segmentation, WDBC, and Sonar data, QGA's average performance is better than CGA's although the best performance of QGA and CGA are identical. The number of features is smaller than of CGA, but the only exception is the case of sonar data. The average and best classification rates of QGA are superior to those of CGA in every respect for the case of Pima-Indians.

As for the average classification rates, QGA is superior to CGA for all experimental data. The performance difference between CGA and QGA is more evident as the number of features becomes larger. In case of Pima-Indians data which have 8 features, the difference between CGA (77.65%) and QGA (77.91%) is only 0.26%. However, in case of Sonar data which have 60 features, the difference between CGA (80.48%) and QGA (84.39%) is 3.9%. Therefore, we could say that the more a data set has features, the better performance it shows.

Ⅳ. Conclusion

QGA is based on the concept and principles of quantum computing such as Q-bits and superposition of state. QGA is known to have an excellent ability of global search due to its diversity caused by the probabilistic representation, and it could approach better solutions than CGA's in a short time.

In this paper, we applied QGA and CGA to compare them for feature selection. The experimental result using UCI data sets shows that QGA is superior to CGA in terms of the classification rates and the number of features on the average. It means that QGA could be used as a promising alternative to CGA for feature selection.

Acknowledgement

This work was supported by Ministry of Culture, Sports and Tourism (MICST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) R&D Program 2011.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0009213)

References

[1] I.-S. Oh, J.-S, Lee, and B.-R Moon, “Hybrid Genetic Algorithms for Feature Selection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, pp. 1424-1437, 2004.

[2] J. Holland, *Adaptation in Nature and Artificial Systems*, MIT Press, 1992.

[3] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Trans. Evolutionary Computation*, Vol. 6, pp. 580-593, 2002.

[4] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ Gate ,and two phase scheme,” *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 2, pp. 156-169, 2004.

[5] T. Hey, “Quantum Computing: an introduction,” *Computing & Control Engineering Journal*, pp. 105-112, 1999.

[6] http://archive.ics.uci.edu/ml/

저자소개

허기수

2000년 전북대학교 기계공학부 산업공학과 학사 졸업.

2004년 전북대학교 정보과학 컴퓨터정보 석사 졸업.

2009년 전북대학교 컴퓨터정보학과 박사 졸업.

2011년~현재: 한국전자통신연구원 선임연구원

<주관심분야 : 영상처리, 패턴인식, 컴퓨터비전>

정현태

1993년 충남대학교 전자공학과 학사 졸업.

1995년 충남대학교 전자공학과 석사 졸업.

1995년~2000년 삼성중공업 중앙연구소 연구원

2001년~현재: 한국전자통신연구원 선임연구원

<주관심분야 : 웨어러블컴퓨팅, 비전기반 사용자 인터페이스>

박아론

2006년 전남대학교 전자컴퓨터정보통신공학부 학사 졸업.

2008년 전남대학교 전자공학과 석사 졸업.

2012년 전남대학교 전자컴퓨터공학과 박사 졸업.

<주관심분야 : 디지털 신호처리, 의료통신 음성관련 신호처리, 패턴인식>

백성준

1989년 서울대학교 전자공학과 학사 졸업.

1992년 서울대학교 전자공학과 석사 졸업.

1999년 서울대학교 전자공학과 박사 졸업.

2002년~현재: 전남대학교 전자컴퓨터공학과 교수

<주관심분야 : 디지털 신호처리, 의료 통신 음성관련 신호처리>