유전자알고리즘을 이용한 탐색공간분할 학습방법에 의한 규칙 생성

장 수 현† · 윤 병 주††

요 약

학습 예(training examples)로부터 규칙을 생성하는 문제는 큰 탐색 공간상에서 많은 지역최소치를 가지고 있는 최적화 문제로 귀결되므로 복잡하고 어려운 문제로 알려져 있다. 이러한 생성규칙을 만들기 위한 여러 가지 학습방법들이 제안되었으며, 그 중 한가지 학습방법이 유전자알고리즘을 연산모델로 사용하는 것이다. 그러나 전통적인 유전자알고리즘은 전역해 부근에서 수렴속도가 떨어지고, 추출된 규칙의 효율성에 문제가 있다. 본 논문에서는 유전자알고리즘의 학습과정에서 포착되는 염색체의 스키마를 분석하여 탐색공간을 부분해(subsolution)를 구할 수 있는 공간들로 분할함으로써, 보다 일반화된 분류 규칙집합을 찾는 방법을 제안하였다. 또한, 실험을 통하여 기존의 기계학습 방법을 사용한 경우와 효율을 상호 비교하여 제안한 방법의 타당성을 입증하였다.

Rule Generation by Search Space Division Learning Method using Genetic Algorithms

Su-Hyun Jang† · Byung-Joo Yoon††

ABSTRACT

The production-rule generation from training examples is a hard problem that has large search space and many local optimal solutions. Many learning methods are proposed for production-rule generation and genetic algorithms is an alternative learning method. However, traditional genetic algorithms has been known to have an obstacle in converging at the global solution area and show poor efficiency of production-rules generated.

In this paper, we propose a production-rule generating method which uses genetic algorithm learning. By analyzing optimal sub-solutions captured by genetic algorithm learning, our method takes advantage of its schema structure and thus generates relatively small rule set.

1. 서 론

지식기반 시스템 개발의 초기에는 해당 분야의 전문가로부터 직접 지식을 획득하여 지식베이스를 구축하였다. 그러나 해당 분야의 전문가로부터 지식을 획득하는 일은 많은 시간과 노력을 필요로 할뿐만 아니라, 만들어진 지식베이스의 정확성과 일관성을 보장할 수 없다. 이와 같이 지식획득 과정에서 야기되는 문제를 지식획득병목현상(knowledge acquisition bottleneck)이라 한다[1,2].

이와 같은 지식획득병목현상을 해결하려는 시도로

† 정 회 원 : 명지대학교 대학원 컴퓨터공학과
†† 종신회원 : 명지대학교 컴퓨터공학과 교수
논문접수 : 1998년 3월 30일, 심사완료 : 1998년 8월 11일

서 기계학습 연구가 시작되었으며, 여러 형태의 학습 방법이 제안되었다[3]. 최근에는 유전자알고리즘도 지식획득을 위한 연산모델로 각광받고 있는데, 유전자알고리즘은 기본적으로 병행탐색을 수행하기 때문에 다른 탐색 전략들보다 최적화 문제를 해결할 높은 가능성을 가지고 있다. 그러나 유전자알고리즘은 최적해 부근으로의 수렴은 빠른 속도로 이행되지만 최적해 부근에서 최적해로의 수렴은 빠른 속도로 이행되지 못한다[4]. 또한, 복잡한 문제 도메인의 규칙집합을 생성하는 문제에서 규칙의 수를 사전에 정의할 수 없기 때문에 학습에 어려움이 있다. 이러한 어려움은 De Jong이 제안한 GABIL시스템[5, 6]과 같이 염색체의 표현방법을 주의 깊게 선택함으로써 부분적으로 해결할 수 있으나, GABIL시스템은 전체 탐색공간에 대한 탐색만을 수행함으로 학습된 결과의 규칙집합은 최적의 일반화된 규칙집합을 생성하기에는 어려움이 있다.

본 논문에서 제안하는 학습 방법은 유전자 알고리즘에 의해 찾아진 부분해(subsolution)가 되는 규칙들 사이의 관계를 나타내는 염색체 구조(genotype schema)를 분석하여 탐색점을 조정하고 탐색 방향을 유도함으로써 최적해로의 수렴속도를 향상시키고, 일반화된 규칙집합을 찾을 가능성을 높이도록 하였다. 또한, 표준 문제로 알려진 MONK 문제와 breast cancer 문제에 제안한 방법을 적용하여 다른 학습 알고리즘들을 적용했을 때와 효율을 비교하며, 생성되는 규칙집합의 크기에 대해서도 고찰한다.

2. GABIL

규칙집합 생성을 위한 유전자 알고리즘의 응용은 크게 두 가지 방향으로 연구되어져 왔다. 그 첫 번째 방법은 유전자 알고리즘의 연산자를 변화시킴으로 탐색의 효율을 높이려는 시도이다[7, 8]. 이러한 시도는 문제 분야의 지식을 이용하여 적합한 유전 연산자를 만들기 때문에 매우 효율적인 탐색을 수행 할 수는 있지만 응용문제의 도메인 지식을 정확하게 나타내 주는 새로운 유전연산자를 만드는데 어려움이 따른다. 두 번째 방법은 전통적인 유전자 알고리즘의 형태에 최소한의 변화로서 유전자 알고리즘의 효율적이고 적응적인 탐색 특성을 유지할 수 있도록 하는 염색체의 표현방법을 만들려는 시도이다[5, 6]. 이러한 시도는 복잡한 규칙집합을 선형 스트링(염색체)에 사상(mapping)하는 방법을 찾기 위해 시스템 설계자가 많은 부담을 안게 되지만, 효과적인 사상방법을 찾게되면 시스템에 약간의 변화를 가함으로써 유전자 알고리즘의 효율적이고 적응적인 탐색 특성을 얻을 수 있다.

본 장에서는 제안한 방법의 염색체 표현을 위한 토대가 되는 GABIL시스템에 대해 먼저 고찰한다. GABIL 시스템은 위의 두 번째 방법을 따르는 대표적인 학습 시스템 중의 하나이다.

2.1 탐색공간 표현

GABIL시스템은 탐색할 규칙집합을 염색체에 표현하기 위해 가변길이 염색체 표현방법을 사용하였다. 이 표현방법은 고정된 길이의 규칙들을 하나의 염색체에 가변 길이로 표현함으로써 규칙집합을 염색체에 사상하는 방법이다.

각 규칙의 표현의미는 다음 예와 같이 설명될 수 있다. 예를 들어, 제시된 학습 예제가 hair, eyes의 특성값으로 설명되고 hair 특성은 {dark, red, blond}, eyes 특성은 {blue, brown}의 순서화된 이산적인 값을 갖는다고 할 때, 한 규칙은 다음과 같은 생성규칙 표현을 가질 수 있다. [IF (hair = red or blond) AND (eyes = blue) THEN class = 0]. 이 규칙 표현에서 조건부는 AND (conjunction)로 이루어져 있고, hair 특성은 내부적인 OR (internal disjunction)로 이루어져 있다. 이와 같은 규칙을 유전자 알고리즘의 염색체에 표현한 형태는 다음과 같다.

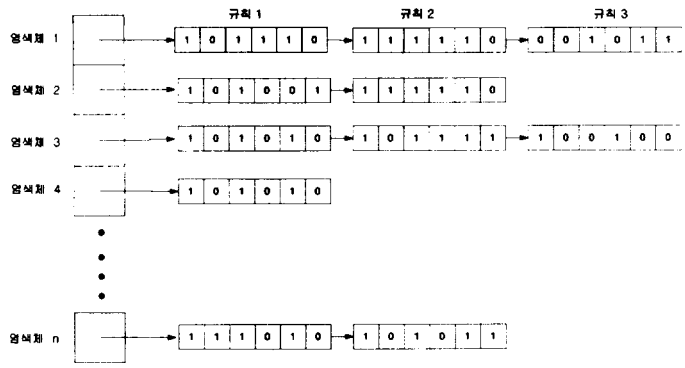
hair eyes class
0 1 1 1 0 0

만약, 모든 특성들이 가질 수 있는 값이 k개라면, 길이 k의 string으로 하나의 규칙을 표현할 수 있다. 그리고 클래스에 대한 표현은 어떤 클래스에 대한 규칙을 생성하기 위해 학습하는 과정에서 생략할 수 있다.

이러한 규칙들로 구성된 규칙집합을 염색체에 표현한 형태는 그림 1과 같다. 이 표현방법은 규칙집합을 한 염색체에 표현하기 때문에 문제 도메인의 각 클래스가 몇 개의 규칙들로 설명이 가능한지 모를 지라도 학습이 가능하다.

2.2 유전 연산자

GABIL시스템은 유전자 알고리즘의 기본연산자인



(그림 20) 규칙집합의 염색체 표현
(Fig. 1) Chromosome representation for rule set

교배연산(crossover)과 돌연변이연산(mutation)만을 수행한다. 교배연산은 가변길이 염색체를 다루는데 용이한 2-point 교배연산을 사용한다. 2-point 교배연산은 그림 2와 같이 두 개의 교차점(crossover site)에서 염색체의 교배를 수행한다. 돌연변이연산은 염색체의 각 비트에 대하여 동일한 확률을 가지고 비트의 값을 바꾼다.

2.3 평가함수(fitness function)

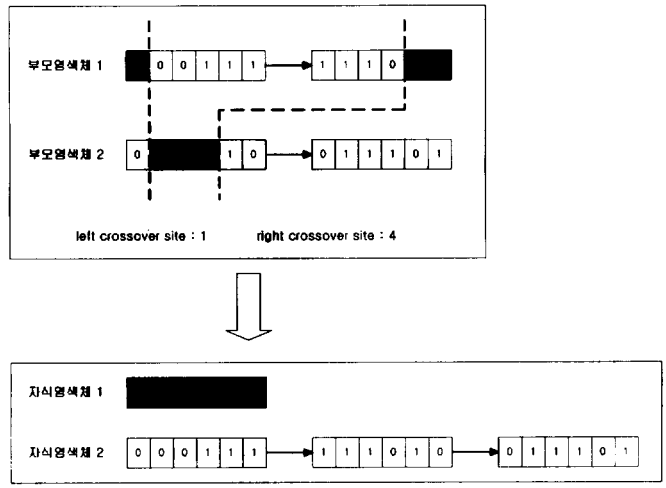
평가 함수는 전체 학습 예에 대하여 정확하게 분류한 예의 비율을 평가함수로 사용하였다. 이를 수식으로 표현하면 식(1)과 같다.

$$F_t(i) = \frac{E_{cp} + E_{nn}}{E_n} \quad (1)$$

단, $F_t(i)$: t 세대에서 염색체 i 의 적합도,
E_{cp} : 염색체 i 에 포함된 양의 예의 개 수,
E_{nn} : 염색체 i 에 포함되지 않은 음의 예의 개 수,
E_n : 전체 예의 개 수.

2.4 학습과정

GABIL은 양의 예와 음의 예를 정확하게 예측할 수 있는 규칙 집합들의 공간을 탐색하기 위해 유전자 알고리즘을 사용한다. GABIL의 학습과정은 그림 3과 같



(그림 2) Two-point 교배연산
(Fig. 2) Two-point crossover operation

이 전통적인 유전자 알고리즘의 학습과 같다. 그러나 GABIL은 규칙집합 학습을 위한 자연스런 표현방법을 제공함으로써 전통적인 유전자 알고리즘의 형태에 최소한의 변화를 가지고 규칙집합 생성 문제를 해결할 수 있음을 보여준다.

```
Begin
t = 0
Initialize Population P(t)
Fitness P(t)
Until (done)
  t = t + 1
  Select P(t) from P(t-1)
  Crossover P(t)
  Mutate P(t)
  Fitness P(t)
end.
```

(그림 3) GABIL 학습 과정
(Fig. 3) GABIL learning procedure

3. 제안한 학습방법

앞장에서 살펴본 GABIL시스템은 복잡한 문제 도메인의 규칙집합 생성 문제 해결을 위해 가변길이 염색체 표현 방법을 제공하고 있다. 그러나 많은 규칙을 생성하여야 하는 복잡한 문제 도메인에서 규칙집합의 탐색공간이 보다 복잡한 형태를 가지게 되므로 수렴속도가 현저하게 떨어지고, 조기 수렴하는 경향을 보이게 된다. 또한 중복되고, 일반화되지 못한 규칙을 생성할 수 있기 때문에 이를 보완하기 위한 여러 가지 방법들이 제안되어졌다. Adaptive GABIL[6]은 Adding, Dropping 연산자를 추가하고 파라메타를 동적으로 변화시키도록 하고 있다. GIL[7]은 문제 분야의 지식(domain-specific knowledge)을 유전연산자로 사용할 수 있도록 하기 위해 AQ[11]시스템의 추론규칙(inference rule)을 결합한 hybrid system 이다.

본 논문에서 제안하는 학습방법은 기본적으로 GABIL의 염색체 표현방법을 사용하고 전역탐색의 과정은 GABIL의 학습방법과 같은 방법을 이용한다. 그러나 GABIL은 전역탐색만 수행하므로 학습의 수렴속도가 떨어지고, 중복된 규칙을 생성할 수 있기 때문에 전역탐색 과정에서 유전자 알고리즘에 의해 찾아진 부분해(subsolution)가 되는 규칙들 사이의 관계를 나타내는 염색체 구조(genotype schema)를 분석하여 하나의 규칙으로 포함할 수 있는 부분 탐색영역들을 설정함으로써 탐색의 효율을 높이고, 중복된 규칙생성의 문제를 해결할 수 있도록 하였다. 이러한 부분 탐색영역의 설정은 스키마 이론(schema theorem)과 직관적으로 일치한다.

3.1 학습과정 개요

규칙집합의 추출은 전역탐색과 지역탐색을 연속적으로 수행함으로써 이루어지는데, 본 논문에서는 전역탐색과 지역탐색을 다음과 같이 정의하여 사용한다.

전역탐색 : 후보 규칙집합을 하나의 염색체에 표현하여 최적의 규칙집합을 찾기 위해 진화하는 규칙집합 레벨의 탐색을 의미한다.

지역탐색 : 부분해의 가능성이 있는 규칙을 찾기 위해 여러개의 부염색체 집단에서 진화하는 규칙 레벨의 탐색을 의미한다.

제안한 학습방법에서 초기의 탐색은 전역탐색을 수행한다. 이 전역탐색 과정은 De Jong[5]의 batch mode GABIL 학습 방법과 같다. 전역탐색 과정에서 학습이 더 이상 진전되지 못하면 현재까지 찾아진 규칙을 근거로 스키마들을 구한다. 스키마는 규칙레벨의 탐색을 위한 탐색공간을 표현한다. 따라서 각 스키마를 이용하여 부분해가 되는 규칙을 찾기 위한 부염색체 집단을 구성하여 각 부염색체 집단에서 부분해를 찾는 과정을 수행하고, 각 부염색체 집단에서 가장 높은 적합도를 갖는 규칙들을 결합하여 전체 학습 예에 대하여 평가한다. 지역탐색 과정은 지역탐색 종료조건을 만나면 지역탐색에서 전역탐색으로 전환한다. 제안한 학습방법의 전체 학습 알고리즘은 그림 4와 같다.

```
t = 0
STEP 1 : / Initialize Population /
   1.1 Initialize Population P(t)
   1.2 Fitness P(t)
STEP 2 : / Global Search Procedure /
   2.1 t = t + 1
```

2.2 Select $P(t)$ from $P(t-1)$
2.3 Crossover $P(t)$ / *2 point crossover* /
2.4 Mutate $P(t)$ /*uniform mutation* /
2.5 Fitness $P(t)$
2.6 If Stoping criterion met, then stop
Else If global search stoping criterion met,
Then go to step 3
Else go to step 2.
STEP 3 : / ***Local Search Procedure*** /
3.1 Get the schema S_i
3.2 Make subpopulation $SP_i(t)$ from S_i
3.3 $t = t + 1$
3.4 For each subpopulation $SP_i(t)$
3.4.1 Select $SP_i(t)$ from $SP_i(t-1)$
3.4.2 Crossover $SP_i(t)$ / *uniform crossover* /
3.4.3 Mutate $SP_i(t)$ / *가변 확율 mutation* /
3.4.4 Fitness $SP_i(t)$
3.5 Fitness $P(t)$ from each subpopulation $SP_i(t)$
3.6 If stoping criterion met, then stop
Else If local search stoping criterion met
Then go to step 2
Else go to step 3.3

(그림 4) 제안한 학습 알고리즘
(Fig. 4) Proposed learning algorithm

3.2 지역탐색 공간 결정

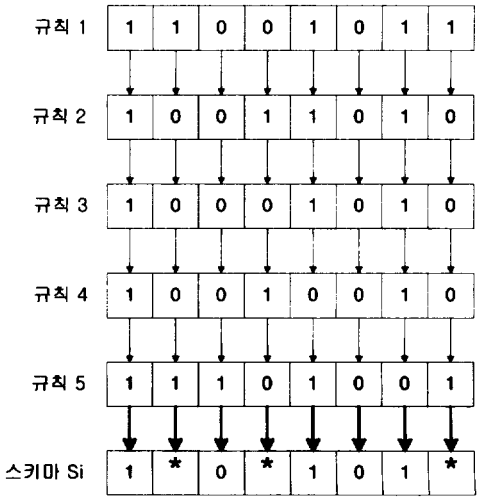
지역탐색의 수행 시기는 전역탐색 과정에서 가장 높은 적합도를 갖는 규칙집합의 적합도가 주어진 세대 이상 향상되지 않게 되었을 때, 여러 개의 축소된 탐색공간에서 각각의 부분해를 찾기 위한 지역탐색으로 전환한다. 축소된 지역탐색 공간의 결정을 위해 전역탐색에서 찾은 규칙집합들 중 상위 20%의 적합도를 갖는 규칙집합에 속하는 규칙들을 사용하고, 각 규칙이 어느 스키마에 속하는 지를 결정하기 위해 규칙이 포함하는 예들을 나타내는 coverage vector를 사용하였다[7]. 규칙들 중 같은 예를 임계치 이상 포함하는 규칙들을 이용하여 하나의 스키마를 구한다. 규칙을 표현한 스트링의 길이를 L이라 할 때, 식(2)에 의해 '0', '1', '*'를 유전인자 값으로 하는 스키마가 구하여진다.

$$S_i = (s_1, s_2, \cdots, s_L) \tag{2}$$

$$s_j = \begin{cases} 1 : \sum_{k=1}^{N} P_{j,k} \geq N \times D_s \\ 0 : \sum_{k=1}^{N} (1 - P_{j,k}) \geq N \times D_s \\ * : otherwise \end{cases}$$

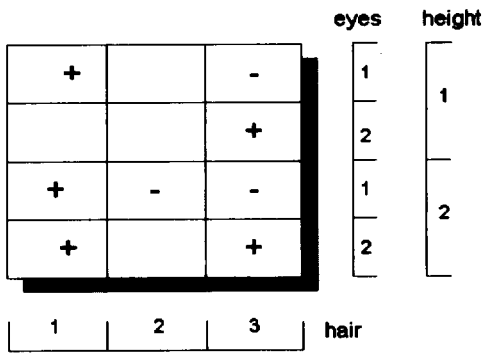
단, s_j : 스키마의 j 번째 비트의 값,
$P_{j,k}$: k 번째 규칙의 j 번째 비트의 값,
N : 공통된 예를 임계치 이상 포함하는 규칙의 수,
D_s : schema detection threshold ($0.5 < D_s \leq 1.0$).

그림 5는 같은 예를 임계치 이상 포함하는 5개의 규칙들로부터 하나의 스키마를 구하는 예이다.

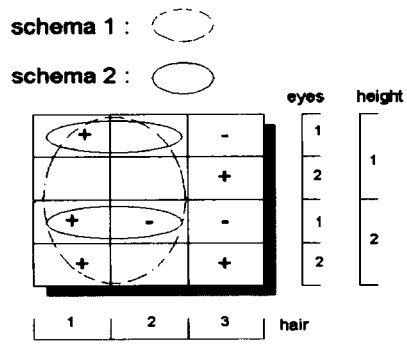


(그림 5) 스키마를 구하는 예 (Ds = 0.8)
(Fig. 5) Example of schema detection

만들어진 스키마들 사이의 관계는 포함 관계를 가질 수 있다. 포함 관계의 스키마들을 합성함으로서 중복된 지역탐색 공간을 줄일 수 있다. 예를 들어, height 특성은 { short, tall }, eyes 특성은 { blue, brown }, hair 특성은 { dark, red, blond } 을 특성 값으로 갖는다고 할 때, 8개의 학습 예가 그림 6과 같이 표현되었다. 포함 관계의 합성은 스키마 1이 "1 * 1 1 * 1 0"이고, 스키마 2가 "1 * 1 0 1 * 0"이라 할 때, 그림 7과 같이 스키마 2의 탐색공간이 스키마 1의 탐색 공간에 포함된다.



(그림 6) 학습예의 가시화된 표현
(Fig. 6) Representation of training example



(그림 7) 스키마 합성의 예
(Fig. 7) Example of schema composition

따라서 이 스키마들을 식(3)에 의해 합성한 결과는 "1 * 1 * * * 0"이 된다.

$$S_k = (s_1, s_2, \cdots, s_L) \qquad (3)$$

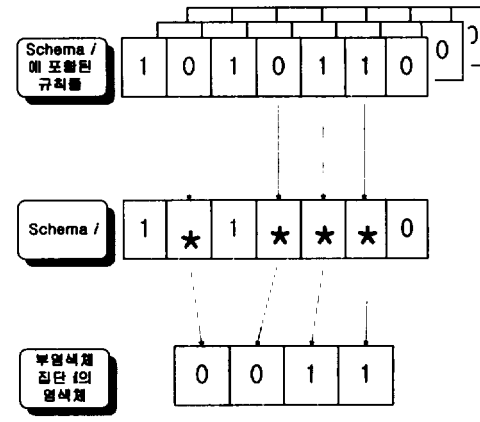
$$s_a = \begin{cases} 1 : S_{a,i} = 1 \wedge S_{a,j} = 1 \\ 0 : S_{a,i} = 0 \wedge S_{a,j} = 0 \\ * : otherwise \end{cases}$$

단, S_a : 스키마의 a 번째 비트 값,
$S_{a,i}$: 스키마 i의 a 번째 비트 값,
$S_{a,j}$: 스키마 j의 a 번째 비트 값.

스키마의 구성과 합성과정은 하나의 규칙으로 포함할 수 있는 합리적인 탐색 공간을 결정할 수 있도록 한다. 또한 각 탐색 공간이 다른 스키마에 의한 탐색 공간과 일치하지 않으므로 중복된 규칙생성을 줄일 수 있다. 그리고 지역탐색을 위해 찾아진 스키마는 최적해의 부분해를 포함할 높은 가능성을 내재하고 있다.

3.3 지역탐색

지역탐색은 스키마 내의 규칙 중 보다 많은 예를 포함하는 최적의 부분해의 가능성을 갖는 규칙을 찾기 위한 탐색이다. 지역탐색을 위한 탐색 공간의 표현은 스키마에 따라 부염색체 집단에 표현된다. 예를 들어, 스키마 i가 "1 * 1 * * * 0"일 때, 스키마에 속하는 규칙 "1 0 1 0 1 1 0"로부터 만들어지는 부염색체 집단, i에 속하는 하나의 염색체 표현은 그림 8과 같다. 부염색체 집단의 염색체 표현은 부염색체 집단에서 찾아야할 규칙(부분해)을 찾는데 있어 탐색공간을 획기적으로 줄이는 역할을 한다.



(그림 8) 부염색체 집단의 염색체 표현
(Fig. 8) Chromosome representation of subpopulation

지역탐색과정에서 사용한 유전연산자는 1-point 교배연산자, 가변확율 돌연변이 연산자를 사용한다. 가변확율 돌연변이 연산자는 축소된 지역탐색 공간을 모두 탐색하기 위해 적합도가 낮을때는 높은 확율을 가지고 적용되고, 적합도가 높아질수록 연산확율이 작아진다. 가변확율 돌연변이 연산의 연산 확율은 식(4)에 의해 결정된다.

$$P_m(t) = MIN_m^{F(t)} \qquad (4)$$

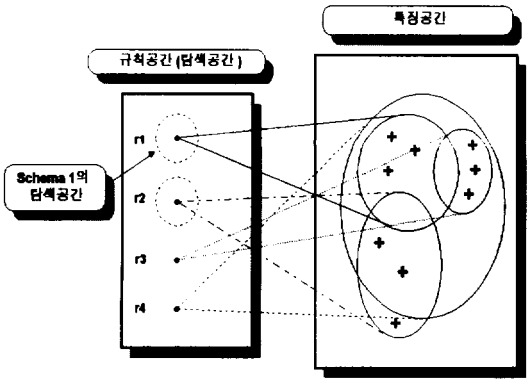
단, $P_m(t)$: t 세대의 돌연변이 연산 확율,
MIN_m : 최소 돌연변이 확율,
$F(t)$: t 세대의 최대 적합도.

각 지역탐색공간에서 염색체의 적합도를 평가하기 위해 전체 학습 예를 분리한다. 지역탐색 공간을 구성하기 위한 스키마에 포함되는 양의 예와 모든 음의 예를 각 부염색체 집단의 학습예로 사용한다. 그리고 여분의 탐색공간으로 부염색체 집단을 하나 더 구성한다. 이 여분의 부염색체 집단의 염색체는 전역탐색에서와 같이 가변길이로 구성하고, 평가를 위한 학습 예는 스키마에 포함되지 않은 양의 예와 모든 음의 예를 가지고 평가한다. 여분의 가변길이 부염색체 집단을 구성함으로써 작은 영향을 미치는 규칙들을 찾을 수 있고, 또한 전체 학습과정에서 염색체의 변화량을 증가시키는 역할을 하게된다.

3.4 지역탐색의 결과로부터 전역탐색을 위한 공간구성

지역탐색 과정에서 찾은 규칙집합의 적합도가 미리 설정된 세대 동안 향상되지 않는다면, 찾아진 스키마가 모든 가능한 부분해들을 찾지 못하였기 때문에 여분의 부분해들을 찾기 위해 다시 전역탐색을 수행할 필요가 있다.

예를 들면, 그림 9에서와 같이 지역탐색으로 찾은



(그림 9) 탐색공간(규칙공간)과 예제공간(특징공간) 사이의 사상

(Fig. 9) Mapping between search space and feature space

규칙 $r1$과 $r2$에 의해 분류할 수 없는 예들을 분류하기 위해 전역탐색에서 규칙집합 { $r1$, $r2$, $r3$ }를 찾거나, 규칙집합 { $r2$, $r4$ }를 찾아야 한다. 이때, 이미 찾아진 규칙 $r1$과 $r2$를 전역탐색을 위한 염색체에 포함하도록 함으로써 최적의 규칙집합을 찾을 가능성을 높일 수 있다. 또한 일반화된 규칙 $r1$과 $r2$를 다시 구성하는데 필요한 노력을 줄인다.

4. 실험 및 고찰

이 장에서는 기계학습 분야에서 표준문제로 사용하고 있는 MONK문제[9]와 breast cancer문제[10]에 제안한 학습방법을 적용한 결과와 기존의 학습방법의 결과를 상호 비교하여 제안한 학습방법의 효율을 평가한다. 비교를 위해 사용한 기존의 학습방법의 결과는 이미 보고된 결과를 사용하였다.

MONK 문제는 서로 다른 6개의 특징들에 의하여 설명되는 로봇의 분류를 위해, 한 종류(class)에 대한 논리적인 설명을 만들어 내도록 하는 인공적(artificial domain)으로 만든 문제이다. 이 문제는 세 가지 서로 다른 수준의 문제를 가지고 있다. MONK-1문제는 평이한 수준의 문제이고, MONK-2는 비교적 많은 규칙에 의해 분류될 수 있는 복잡한 문제이다. 또한 MONK-3는 학습 예에 대하여 5% 수준의 클래스 잡음(class noise)이 있는 문제이다.

breast cancer 문제는 기계학습 분야에서 많이 사용되는 실세계(natural domain)문제 중 하나 이다. 이 문제는 5년동안 어떤 병원에서 검진받은 286명의 여성 환자들의 진료기록 데이터로부터 만들어 졌다. 각 예들은 9개의 특성들로 설명되고, 각 특성이 가질 수 있는 값은 평균 5.8개이다. breast cancer 문제에는 완전히 일치하는 특성값을 가지고 있는 두 예가 서로 다른 클래스로 분류된 클래스 잡음(class noise)이 있다. 또한 이 데이터에는 9개의 예에서 특성값이 빠진(missing attribute value) 상태로 존재한다.

4.1 MONK 문제에서의 결과 분석

제안한 학습방법으로 MONK 문제를 학습하기 위해 사용된 실험환경은 다음과 같다. 3개의 문제에 대하여 공통적으로 염색체의 수는 30개로 하였고, 염색체의 교배확률은 70%, 돌연변이 연산자 적용확률은 3%, 최대 세대수는 5000세대, 초기 염색체는 3개의 규칙을 갖도록 초기화하였다. MONK문제는 432개의 가능한 예들 중 MONK-1은 124, MONK-2는 169, MONK-3은 122개의 예를 학습 예로 사용하였다. 이는 다른 학습방법들에서도 공통적으로 사용한다.

각 학습 문제에 대하여 10회 학습하여 테스트한 새로운 예(unseen examples)의 클래스를 예측한 결과는 표 1과 같다. 그리고 학습한 결과 만들어진 규칙집합의 크기는 표 2와 같다. 비교된 학습방법들의 결과는

기존의 보고된 결과를 참조한 것이다[9].

비교된 학습방법들에 대한 간단한 설명은 다음과 같다. ID3[12]는 각 노드에서 학습 예들을 가장 잘 분류하는(Information gain이 가장 큰) 특징(attribute)을 반복적으로 찾아 트리를 구성하는 Top-down 방식의 결정트리(decision tree) 생성방법이다. ID5R[13]은 ID3를 점진적 학습(Incremental learning)이 가능하게 확장한 학습 방법이다. CN2[14]는 ID3와 AQ알고리즘의 장점을 결합한 형태로, 학습 결과는 AQ와 같이 If-Then 규칙의 순서화된 리스트로 표현되지만, 규칙은 ID3와 같이 Top-Down 방식으로 조사된다. PRISM[15]은 ID3알고리즘을 기초로 한다. 그러나, ID3가 attribute-value 쌍의 평균 엔트로피(entropy)를 최소화하려고 하는 반면, PRISM은 attribute의 특성 값들 중 분류를 위해 많은 정보를 제공하는 특성 값을 찾는 알고리즘이다. mFOIL[16]은 학습 예들 사이의 관계를 학습하는 시스템으로 학습 예들과 background knowledge로부터 논리 프로그램(logic program)들의 집합을 생성하는 알고리즘이다. Assistant Professional[17]은 ID3에 기초한 학습 알고리즘이다. 그러나, 특성 값을 이진화 하여 보다 작고 정확한 결정트리를 생성하고, 트리를 전지(pruning)하는 방법을 가지고 있어 잡음이 있는 학습 예에 의한 overfitting 현상을 제거한 학습 알고리즘이다. AQ17-DCI[9]와 AQ17-HCI[9], AQ15-GA [18]는 AQ알고리즘을 확장한 것들이다. AQ17-DCI는 두 개의 과정을 거치게 되는데, 첫 번째 과정은 AQ알고리즘의 STAR알고리즘과 유사하게 규칙을 생성하고, 두 번째 과정은 생성된 규칙을 특수화(specialization)하면서 새로운 특성을 만든다. AQ17-HCI는 현재 만든 가설(hypothesis)을 분석하여 다음 학습과정에서 새로운 특성을 만드는 알고리즘이다. AQ15-GA는 유전자 알고리즘을 이용하여 특징 선택을 하는 hybrid 학습방법이다.

제안한 방법에 의한 예측성능은 MONK-1문제에 대하여 10번의 실험 중 8번은 100%의 예측성능을 보였다. 2번은 필요 이상의 규칙을 생성하여 일반화가 충분히 되지 못하였기 때문에 완전한 예측 성능을 보이지 못하였다. MONK-1문제에 대한 예측성능의 결과는 대부분의 다른 학습방법과 유사한 예측성능을 보였다. MONK-2문제에 대하여 제안한 방법이 AQ류의 학습 알고리즘을 제외한 대부분의 학습 알고리즘보다 우수한 성능을 보였다. 특히MONK-2문제에 제안한 방법은 10번의 실험 중 7번은 학습이 수렴되었고, batch mode GABIL은 한번도 수렴을 하지 못하였다. 이는 batch mode GABIL의 학습은 영향력이 큰 특정 규칙만을 생성하는 방향으로 학습이 진행되어 조기수렴 되었기 때문이다. 그러나 제안한 학습방법에서는 지역탐색의 과정에서 전역탐색의 조기수렴된 결과를 유지하면서 동시에 염색체 집단의 변화량을 주어 다른 영역의 규칙을 찾을 수 있는 방법을 가지고 있기 때문에 작은 영향력을 가지는 규칙도 생성할 수 있었다. MONK-3문제에 대하여 제안한 학습방법은 대부분의 학습방법보다 우수한 성능을 보이고 있다. 이는 잡음이 있는 문제에 대하여 제안한 학습방법이 강력한 일반화성능을 가지고 있음을 보여준다.

〈표 1〉 MONK 문제에서의 예측 정확도 (prediction accuracy) 비교
〈Table 1〉 Comparison of prediction accuracy in the MONK problem set

학습 방법		예측 정확도 (%)		
		MONK-1	MONK-2	MONK-3
ID3		83.2	69.1	95.6
ID5R		79.7	69.2	95.2
CN2		100	69.0	89.1
PRISM		86.3	72.7	90.3
mFOIL		100	69.2	100
Assistant Professional		100	81.3	100
AQ17-DCI		100	100	94.2
AQ17-HCI		100	93.1	100
AQ15-GA		100	86.8	100
Backpropagation		100	100	93.1
batch mode GABIL		86.5	71.8	96.8
제안한 방법	**best**	**100**	**90.5**	**100**
	average	**99.2**	**88.9**	**97.3**

제안한 방법에 의해 생성된 규칙집합의 크기는 일반적으로 생성될 수 있는 최적의 규칙에 가깝게 생성되었다. MONK-1문제의 경우 생성될 수 있는 최적의 규칙은 4개이다. 제안한 방법의 경우 평균 5.8개의 규칙을 생성하였는데 반해 대부분의 방법들이 필요이상의 많은 규칙을 생성하여 충분한 일반화 성능을 보이지 못하였다. MONK-2 문제는 최적의 규칙 수가 15개이다. 대부분의 학습방법의 경우 최적의 규칙 수 보다 상당히 많은 규칙을 생성하였다. 반면 제안한 방법은

최적의 규칙 수와 유사한 18.4개의 규칙들을 만들 수 있었다. batch mode GABIL 에서는 평균 6.1개의 규칙 집합을 만들었다. 그 이유는 앞에서도 설명했듯이 영향력이 큰 특정 규칙만을 생성하는 방향으로 학습이 진행되었기 때문이다. MONK-3문제는 완전하지 못한 학습 예를 가지고 있다. 그리고 최적의 규칙집합은 2개의 규칙으로 구성될 수 있다. 이 문제에 대하여 제안한 방법과 batch mode GABIL 모두 평균 2.3개의 규칙을 생성하였다.

〈표 2〉 MONK 문제에서의 규칙집합 크기 비교
(괄호 안의 숫자는 양의 예에 대한 규칙 수와 음의 예에 대한 규칙 수를 의미함)

〈Table 2〉 Comparison of the size of rule set in the MONK problem (The values in parentheses indicate the size of positive and negative rules set, resp.)

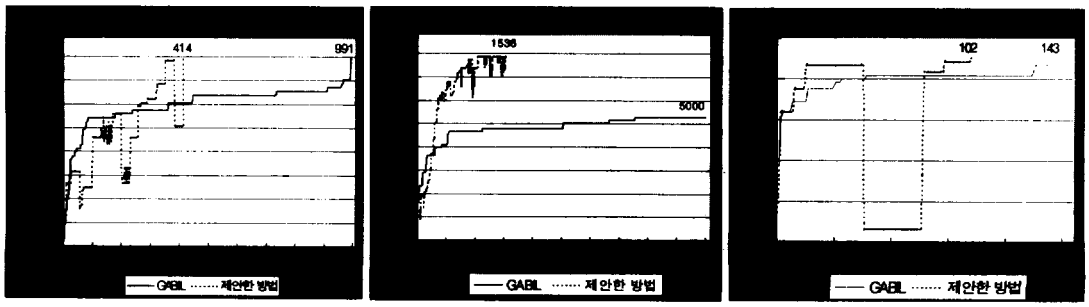
학습 방법		규칙집합 크기		
		MONK-1	MONK-2	MONK-3
ID3		62	110	31
ID5R		52	99	28
CN2		10	58	24
PRISM		29	73	26
mFOIL		4	19	2
Assistant Professional		8	57	5
AQ17-DCI		5(4, 1)	17(15, 2)	7(2, 5)
AQ17-HCI		7(4, 3)	16(9, 7)	7(2, 5)
AQ15-GA		-	-	7(2, 5)
batch mode GABIL		6.2	6.1	2.3
제안한 방법	best	4	17	2
	average	5.8	18.4	2.3

표 1과 표 2의 결과를 살펴보면, 제안한 방법은 AQ류의 학습방법을 제외한 대부분의 학습알고리즘과 유사하거나 좋은 성능을 보이고 있다. AQ류의 학습알고리즘들은 학습과정에서 문제분야의 지식(domain-specific knowledge)을 background knowledge로 사용하여 여러개의 후보규칙들을 생성하고, 선호 기준(preference criterion)에 가장 적합한 규칙을 하나하나 반복적으로 만들기 때문에 예측 정확도의 측면에서는 우수하나 상당한 양의 연산 복잡도가 필요한 학습기법이다.

그림 10은 MONK 문제들에서 제안한 방법과 batch mode GABIL의 수렴과정을 보여준다. 수렴과정 중 적합도 가 급격히 떨어지는 세대가 있다. 그 원인은 전역탐색의 과정에서 탐색이 더 이상 수렴되지 않는 지역 최소치에 빠져 있다고 가정하고 지역탐색을 수행하게 될 때 염색체 집단에 변화량을 주었기 때문이다. 제안한 학습방법의 수렴속도는 지역최소치에서 빨리 빠져 나올 수 있도록 적절한 시기에 염색체집단의 변화량을 주었기 때문에 GABIL 방법보다 빠르게 전역해에 수렴한다.

3.2 breast cancer 문제에서의 결과 분석

제안한 학습방법으로 breast cancer 문제를 학습하기 위해 사용된 실험환경은 다음과 같다. 염색체 집단내 염색체의 수는 30개로 하였고, 염색체의 교배확률은 70%, 돌연변이 연산 최소 적용확률은 3%, 초기 염색체는 3개의 규칙을 갖도록 초기화하였다. Breast cancer 문제는 학습(training)데이터와 검증(test)데이터가 따로 존재하지 않는 문제이므로 4 fold cross validation 방법으로 평가를 하였다. 제안한 방법을 다



① MONK-1 수렴도 ② MONK-2 수렴도 ③ MONK-3 수렴도

(그림 10) MONK 문제에서 제안한 방법과 GABIL의 수렴 비교

(Fig. 10) Comparison of convergency in the MONK problems with GABIL *vs* proposed method

른 학습방법과 비교하기 위하여 기존의 보고된 결과들과 비교하였다[6, 7, 10].

비교한 결과는 표 3과 같다. 표 3의 결과는 제안한 방법의 예측성능이 back propagation에 의한 신경망 학습방법과 불완전한 예들을 위해 backtracking을 하여 트리를 전지하는 학습방법인 Assistant professional /with tree pruning을 제외한 대부분의 학습 방법들 보다 비교적 우수한 성능을 보이고 있다. back propagation 학습방법은 breast cancer 문제에서 비교적 우수한 성능을 보이지만 학습된 지식이 노드사이의 연결강도로 표현되기 때문에 이를 규칙형태로 만들기 위한 방법을 필요로 한다.

〈표 3〉 breast cancer 문제에서의 학습결과 비교
〈Table 3〉 Comparison of learning result in the breast cancer problem set

학습 방법	예측 정확도(%)	규칙집합 크기
Human expert	64	-
AQ 15	68	41
Assistant professional /without tree pruning	67	63(leaves)
Assistant professional /with tree pruning	72	9(leaves)
ID5R	63.4	-
Backpropagation	71.5	-
GABIL	68.7	43
Adaptive GABIL	70.3	-
GIL	65	36
GIL / with emphasized cost	67	10
제안한 방법	70.4	15.15

규칙집합의 크기측면에서 제안한 방법은 Assistant professional / with tree pruning과 규칙집합의 평가에서 cost(complexity)의 영향을 크게 함으로써 잡음이 있는 데이터에 대하여 과도한 학습(overfitting)을 하지 못하도록 한 GIL/with emphasized cost 방법을 제외한 대부분의 학습방법들보다 작은 규칙들을 만들었다.

이 실험 결과는 제안한 학습방법이 실세계 문제에서도 다른 기계학습 방법보다 우수한 성능을 보일 수 있음을 보여준다. 따라서, 실세계 문제의 해결을 위한 지식베이스 시스템의 부분으로서 지식베이스의 구축을 위한 방법으로 사용될 수 있다. 또한 간결한 규칙을 생성하기 때문에 이해하기 어려운 문제 분야에 적용함으로서 사람들이 문제 분야의 특성을 이해하기 쉬운 규칙을 제공할 수 있다.

5. 결 론

본 논문에서는 유전자 알고리즘을 이용한 효율적인 규칙집합 생성 방법을 제안하였다. 제안한 학습방법은 유전자 알고리즘이 어떻게 전역해를 찾는 지를 효과적으로 보여주는 스키마 이론에 근거하여 탐색점을 조정하고 탐색 방향을 유도할 수 있었다.

실험을 통하여 제안한 학습방법의 효율성을 다른 학습방법들과 비교한 결과 제안한 방법은 기존의 다른 방법들 보다 비교적 우수한 예측의 정확성을 보였고 생성한 규칙집합의 크기 측면에서도 평균적으로 우수하였다. 또한 여러 문제들에 대하여 같은 파라메타를 사용하여 학습하였음에도 특정한 문제에 대하여 성능이 크게 저하되지 않았다. 이는 문제 분야의 지식(domain-specific knowledge)을 부가적으로 사용하지 않고 여러 형태의 문제에 제안한 학습방법을 사용할 수 있음을 의미한다.

그러나, 제안한 방법은 이산적인 특성값을 갖는 학습 예에 대한 학습을 전제로 하고 있다. 따라서 연속적인 특성값을 갖는 문제를 제안한 학습방법에 적용하기 위해서는 연속적인 값을 갖는 특성값을 이산적인 값으로 변환해 주어야 한다. 이는 실세계의 문제들이 대부분 연속적인 값을 갖는 문제라는 점을 감안하면 커다란 제약이라 할 수 있다. 따라서 앞으로의 연구는 연속적인 값을 갖는 문제에 제안한 학습방법을 적용하기 위한 방법에 연구의 촛점이 모아져야 할 것이다.

참 고 문 헌

[1] Sabrina Sestito and Tharam S. Dillon, *Automated Knowledge Acquisition*, Prentice-Hall, 1994.

[2] B. G. Buchanan and D. C. Wilkins(eds.), *Readings in Knowledge Acquisition and Learning*, Morgan Kaufmann, 1993.

[3] Jude W. Shavlik and Thomas G. Dietterich (eds.), *Readings in Machine Learning*, Morgan Kaufmann, 1990.

[4] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Leaning*, Addison-Wesley, 1989.

[5] Kenneth A. De Jong and William M. Spears, "Learning Concept Classification Rule using Genetic Algorithms," IJCAI '91, Vol.2, pp.651-656, Morgan Kaufmann, 1991.

[6] Kenneth A. De Jong, William M. Spears and Diana F. Gordon, "Using Genetic Algorithms for Concept Learning," Machine Learning, Vol.13. pp.161-188, Kluwer Academic, 1993.

[7] C. Z. Janikow, "A Knowledge-Intensive Genetic Algorithm for Supervised Learning," Machine Learning, Vol.13, pp.189-228, Kluwer Academic, 1993.

[8] J. R. Koza, "Concept Formation and Decision Tree Induction using Genetic Programming Paradigm," in *Parallel Problem Sloving from Nature*, pp.124-128, Springer-Verlag, 1991.

[9] S. B. Thrun, et al., "The Monk's Problems : A Performance Comparison of Different Learning Algorithms," CMU-CS-91-197, 1991.

[10] Sholom M. Weiss and Ioannis Kapouleas, "An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods," in *Readings in Machine Learning*, Jude W. Shavlik and Thomas G. Dietterich (eds.), pp.177-183, Morgan Kaufmann, 1990.

[11] R. S. Michalski, "A Theory and Methodology of Inductive Learning," in *Readings in Knowledge Acquisition and Learning*, B. G. Buchanan and D. C. Wilkins(eds.), pp.323-348, Morgan Kaufmann, 1993.

[12] J. R. Quinlan, "Induction of Decision Trees," in *Readings in Machine Learning*, Jude W. Shavlik and Thomas G. Dietterich(eds.), pp.57-69, Morgan Kaufmann, 1990.

[13] P. E. Utgoff, "An Improved Algorithm for Incremental Induction of Decision Trees," International Conference on Machine Learning, pp. 318-325, Morgan Kaufmann, 1994.

[14] P. Clark and T. Niblett, "The CN2 Induction Algorithm," Machine Learning 3, pp.261-283, Kluwer Academic, 1989.

[15] J. Cendrowska, "PRISM : An algorithm for inducing modular rules," in *Knowledge Acquisition for Knowledge-Based Systems*, Gaines and Boose(eds.), pp.253-274, Academic Press, 1988.

[16] J. R. Quinlan, "Determinate Literals in Inductive Logic Programming," IJCAI '91, pp.746-750, 1991.

[17] B. Cestnik, I. Kononenko and I. Bratko, "ASSISTANT 86 : A Knowledge-elicitation Tools for Sophisticated Users," in *Progress in Machine Learning*, Bratko and Lavloc (eds.), Sigma Press, 1987.

[18] H. Vafaie and K. A. De Jong, "Improving A Rule Induction System Using Genetic Algorithms," Machine Learning 4, pp.453-469, Morgan Kaufmann, 1994.

장 수 현

shjang@ce.myongji.ac.kr

1993년 명지대학교 전자계산학과 (공학사)

1996년 명지대학교 대학원 컴퓨터공학과(공학석사)

1997년~현재 명지대학교 대학원 컴퓨터공학과 박사과정

관심분야 : Machine Learning, Knowledge-Based System, Artificial Life

윤 병 주

yoonbj@wh.myongji.ac.kr

1975년 경북대학교 수학과(학사)

1982년 한국과학기술원 전산학과 (석사)

1994년 Florida State University 전산학과(박사)

1977년~1979년 KAL 시스템개발실 근무

1982년~현재 명지대학교 컴퓨터공학과 교수

관심분야 : Machine Learning, Knowledge-Based System, Hybrid Intelligent Systems