분류기 앙상블 선택을 위한 혼합 유전 알고리즘

김 영 원[†] · 오 일 석[††]

요 약

이 논문은 최적의 분류기 앙상블 선택을 위한 혼합 유전 알고리즘을 제안한다. 혼합 유전 알고리즘은 단순 유전알고리즘의 미세 조정력을 보완하기 위해 지역 탐색 연산을 추가한 것이다. 혼합 유전 알고리즘의 우수성을 입증하기 위해 단순 유전 알고리즘과 혼합 유전 알고리즘 각각을 비교 실험하였다. 또한 혼합 유전 알고리즘의 지역 탐색 연산으로 두 가지 방법(SSO: 순차 탐색 연산, CSO: 조합 탐색 연산)을 제안한다. 비교 실험 결과는 혼합 유전 알고리즘이 단순 유전 알고리즘에 비해 해를 탐색하는 능력이 우수하였다. 또한 분류기들의 상관관계를 고려한 CSO 방법이 SSO 방법보다 더 우수하였다.

키워드 : 분류기 앙상블, 분류기 선택, 혼합 유전 알고리즘

Hybrid Genetic Algorithm for Classifier Ensemble Selection

Young-Won Kim[†] · Il-Seok Oh[††]

ABSTRACT

This paper proposes a hybrid genetic algorithm (HGA) for the classifier ensemble selection. HGA is added a local search operation for increasing the fine-turning of local area. This paper apply hybrid and simple genetic algorithms (SGA) to the classifier ensemble selection problem in order to show the superiority of HGA.

And this paper propose two methods (SSO: Sequential Search Operations, CSO: Combinational Search Operations) of local search operation of hybrid genetic algorithm. Experimental results show that the HGA has better searching capability than SGA. The experiments show that the CSO considering the correlation among classifiers is better than the SSO.

Key Words : Classifier Ensemble, Classifier Selection, Hybrid Genetic Algorithm

1. 서 론

패턴인식 시스템의 성능 향상을 위한 방법으로 여러 개의 분류기를 결합하는 연구가 있고, 많은 논문들이 실험적으로 인식률 향상에 효과적임을 입증하였다 [1-4].

분류기 결합에서 주요 관심사는 다음과 같다 [8].

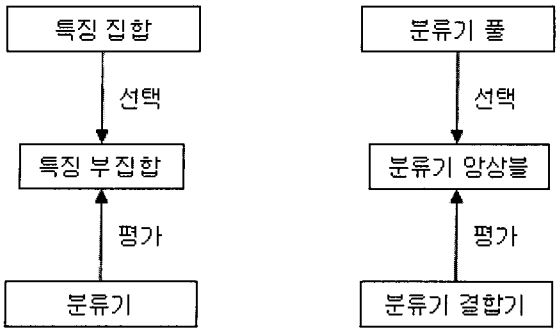
분류기 풀(classifier pool)을 어떻게 생성할 것인가?
분류기 풀에서 어떤 분류기들을 선택할 것인가?
분류기들의 결과들을 어떻게 결합할 것인가?

먼저, 분류기 풀의 생성할 때 고려할 문제는 분류기 풀의 다양성(diversity)이다. 분류기 앙상블은 하나 이상의 분류기가 서로 도와 성능을 향상시키고자 하므로, 분류기들이 에러에 대한 경향이 서로 다를수록 유리하다. 분류기 풀을 만드는 방법으로는 훈련 데이터 집합으로 부터 여러 부집합을 만들어 사용하는 Bagging[5,6]과, Boosting[7], 그리고 특징 집합을 여러 부집합으로 만들어 사용하는 방법[8-10]이 있다.

Ho는 분류기 풀은 만들어져 있다는 가정 하에 분류기의 선택 문제와 분류기 결과의 결합 문제를 각각 범위 최적화(coverage optimization)와 결정 최적화(decision optimization)로 구분하여 이름을 붙였다[8].

분류기 앙상블 분야의 연구자들은 결정 최적화 문제에 많은 노력을 기울여왔다[11-14]. 그러나 범위 최적화[8, 15]로 만들어진 분류기 앙상블의 출력이 바로 결정 최적화 알고리즘의 입력이 되므로 범위 최적화에 대한 연구 또한 매우 중요하다. 범위 최적화는 N 개의 분류기를 갖는 분류기 풀(classifier pool)에서 d 개의 분류기를 갖는 분류기 부집합을 선택하는 문제를 다룬다. 이렇게 선택된 분류기 부집합을

※ 본 연구는 한국과학재단 특정기초연구(R01-2003-000-10879-0)의 지원으로 수행되었습니다.
† 정 회 원 : 한국전자통신연구원 우정기술연구센터 연구원
†† 정 회 원 : 전북대학교 전자정보공학부 교수
논문접수 : 2007년 5월 29일, 심사완료 : 2007년 8월 3일



(그림 1) 특징 선택 문제와 분류기 앙상블 선택 문제의 유사성

분류기 앙상블(classifier ensemble)이라 부른다.

본 논문에서는 분류기 앙상블을 위한 범위 최적화 (분류기 선택 문제)에 초점을 둔다. 범위 최적화의 목적은 인식 성능 측면에서 최적의 분류기 앙상블(optimal classifier ensemble)을 찾는 것이다. 탐색 공간의 크기가 ${}_N C_d$ 이므로 모든 후보 해를 탐색해보는 방법은 최적의 해를 보장하나 N이 커질수록 시간비용이 지수적으로 커지는 문제를 안고 있다. 따라서 주어진 시간 내에 높은 품질의 부최적해 (suboptimal solution)를 찾는 알고리즘이 필요하다.

(그림 1)에서 알 수 있듯이 특징 선택 문제와 분류기 앙상블 선택 문제는 서로 다른 구성 요소를 가지고 있지만 전체적인 구조는 같다. Oh 등은 특징 선택 문제를 해결하기 위해 혼합 유전 알고리즘(HGA: Hybrid Genetic Algorithm)을 제안하였고, 단순 유전 알고리즘과 순차 탐색 알고리즘과 같은 기존 알고리즘들에 비해 탐색 능력이 우수함을 실험적으로 증명하였다[16].

이 논문은 분류기 앙상블 선택을 위해 혼합 유전 알고리즘을 제안하였다. 본 논문의 주요 목적은 분류기 앙상블 선택에 있어서 혼합 유전 알고리즘이 단순 유전 알고리즘에 비해 우수한 탐색 능력을 가지고 있음을 증명하는 것이다. 객관적인 실험을 위해 다양한 특성을 갖는 표준 데이터 집합을 사용하였으며 탐색 공간 크기를 다양하게 설정하였다. 실험 결과는 혼합 유전 알고리즘이 단순 유전 알고리즘에 비해 해의 탐색 능력과 안정성 면에서 우수하였다.

2. 분류기 앙상블 선택을 위한 유전 알고리즘

분류기 앙상블 선택 문제란 N 개의 분류기 중에 최적의 d 개의 분류기 부집합을 선택하는 문제이다. 성능과 오류 경향이 다른 N 개의 분류기는 이미 생성되어 있다고 가정하고 결합 방법은 고정하였다. 유전 알고리즘을 이용하여 찾고자 하는 최적의 해는 결합 인식기의 인식률이 최고가 되는 때의 분류기 앙상블이다.

Zhou 등은 분류기 앙상블 선택을 위해 단순 유전 알고리즘을 이용하였다 [2,3,21]. (그림 2)는 분류기 앙상블 선택을 위한 유전 알고리즘의 구조이다. k는 다음 세대로 진화할 때 대치되는 자식해(offspring)의 수이다. k가 해집단의 크기와 같을 때는 세대형 (generational) 유전 알고리즘이 되고, k가 1일 때는 안정상태 (steady-state) 유전 알고리즘이 된다.

```
초기 해집단 생성;

repeat {
  for i=1 to k {
     두 염색체 p1, p2 선택;
     offspringi = crossover(p1, p2);
     offspringi = mutation(offspringi);
     if (HYBRID) local_search(offspringi);
  }
  offspring1, ..., offspringk를 대치;
} until (정지 조건 만족);
현재까지 최상의 염색체를 return;
```

(그림 2) 유전 알고리즘의 전체 구조

2.1 단순 유전 알고리즘 (SGA: Simple Genetic Algorithm)

다음 세대로의 진화 방법은 k=1인 안정형을 사용하였다. 유전 알고리즘은 매개 변수 설정에 따라 해 품질이 크게 영향을 받으므로, 가능한 자세히 알고리즘과 매개변수에 대해 기술한다. 단순 유전 알고리즘에서는 (그림 2)의 불린 변수 HYBRID를 FALSE로 설정하여 local_search() 연산을 수행하지 않는다. local_search() 연산은 지역 탐색 연산으로 유전 알고리즘의 취약점인 미세 조정력을 보완하기 위한 것이다.

2.1.1 염색체 표현

염색체의 길이는 분류기 풀의 크기인 N이다. 염색체는 이진열(binary string)로 표현되며 '1'은 분류기가 선택되었음을 의미하고 '0'은 배제되었음을 의미한다. 따라서 '1'인 유전자의 개수는 선택된 분류기의 개수인 d와 같아야 한다. 예를 들어 하나의 해가 1000100101 (N=10인 경우) 일 때, 1번, 5번, 8번, 10번 분류기가 선택되어 있고 분류기 앙상블 크기는 4이다.

2.1.2 초기해 집단

각 염색체는 임의수를 생성하여 d 개의 유전자가 '1'의 값을 갖도록 초기화 한다. 해집단(population)의 크기는 20으로 한다.

2.1.3 적합도 계산

분류기 앙상블에 속한 분류기들에 대해 투표 (voting) 방법에 의해 결합 인식률을 구하고 이를 이 해의 품질로 한다. 투표 결과가 동률일 때는 동률인 투표 집단에서 가장 우수한 분류기가 들어있는 투표 집단의 인덱스로 결정하였다. 이 품질을 아래에서 설명하는 순위기반 방법을 이용하여 적합도(fitness)로 변환하였다. 즉 해집단 내의 해들을 품질(결합 인식률)에 따라 정렬한 후, i 번째 해의 적합도를 식 1에 따라 배정한다. 결국 가장 좋은 해는 max 값, 가장 나쁜 해

는 min 값을 갖게 되고, 모든 해는 그 사이 값을 갖게 된다. min과 max값의 차이가 클수록 선택압(selection pressure)이 높아진다. 우리는 min=0.1, max=1.0을 사용하였다.

$$f_i = \max + (i-1) \times (\min - \max)/(N-1), \quad 1 \le i \le N \quad (식1)$$

2.1.4 선택

룰렛-휠 방법에 의해 두 개의 부모 염색체를 선택한다.

2.1.5 교배와 돌연변이

n개의 자름선을 임의로 선택한 후, 두 부모 염색체를 교차 시켜 자식 염색체를 생성한다. n은 3으로 하였다. 그 다음 두 개의 자식 염색체는 돌연변이 연산을 거친다. 이때 염색체의 모든 유전자에 대해 [0,1] 사이에 있는 임의의 수 r을 생성하여 r이 돌연변이 확률(P_m=0.1)보다 작은 경우에는 해당 유전자의 값을 0은 1로 1은 0으로 변환하였다. 돌연 변이 과정은 임의수 r에 의존하므로 연산이 끝난 후 '1'인 유전자의 수가 d와 항상 일치하지는 않는다. 따라서 돌연 변이 후, '1'인 유전자의 수가 d와 다를 경우 유전자를 임의로 선택하여 1-0 또는 0-1 변환을 시켜 '1'인 유전자의 수가 d가 되도록 하였다.

2.1.6 대치

두개의 자식 중에 우수한 자식을 선택하여 두 부모보다 우월하면 자신과 비슷한 부모를 대치하고, 두 부모 사이라면 열세한 부모를 대치한다. 자식이 두 부모보다 열세하다면 해 집단에서 가장 열세한 염색체와 대치한다.

2.1.7 종료

유전 과정의 세대 수가 미리 정한 최대 세대 수 T에 도달하면 중단한다.

2.2 혼합 유전 알고리즘 (HGA: Hybrid Genetic Algorithm)

유전 알고리즘은 교배나 변이 연산을 통해 지역 최적해에 빠지는 것을 방지할 수 있고, 선택 압력을 적절하게 조절하여 문제 공간의 넓은 범위를 탐색할 수 있다. 그러나 유전 알고리즘은 지역 최적해 근처에서 미세 조정력이 약하기 때문에 긴 실행 시간을 필요로 한다. 이러한 이유로 특징선택 문제[16], TSP 문제[17], 그래프 분할 문제[18],영상 압축 문제[19] 등 많은 응용에서 단순 유전 알고리즘에 미세 조정력을 향상시키기 위해 혼합 유전 알고리즘을 적용하였다[16]. 교배와 변이 연산을 통해 자식 염색체는 부모 염색체로부터 좋은 특성들을 상속 받는다. 이렇게 생성된 자식 염색체는 그들의 부모보다 우수하거나 열세할 수 있다. 혼합 유전 알고리즘은 자식 염색체들을 대치 연산 전에 지역 탐색 연산을 통해서 품질을 향상 시킨다.

지역 탐색 연산의 중요 관점은 지역 향상을 위해 적절한 연산을 선택하는 것이다. 이 논문에서 제안하는 지역 탐색 연산은 두 가지이다. 첫 번째 방법은 한 번에 하나의 유전자를 고려하는 순차 탐색 (sequential search) 연산을 사용한다. 두 번째 방법은 한 번에 여러 개의 유전자를 같이 고려하는 조합 탐색 (combinational search) 연산을 사용한다. 두 개의 지역 탐색 연산의 목적은 '1'인 유전자의 수가 d와 같도록 수정하는 것과 동시에 자식 염색체의 품질을 향상시키는 것이다.

2.2.1 순차 탐색 연산(SSO: Sequential Search Operations)

이 방법이 사용하는 지역 탐색 연산의 구조는 아래와 같다.

```
sequential_local_search(offspring) {
  count = offspring의 '1'인 유전자의 수;
  switch {
     case count=d : RippleRem(r); RippleAdd(r);
     case count<d : for(i=0; i<d-count; i++) RippleAdd(r);
     case count>d : for(i=0; i<count-d; i++) RippleRem(r);
  }
}
```

RippleRem(r)과 RippleAdd(r)은 특징 선택 문제를 위해 Oh 등에 의해서 처음 제안되었다 [16].

rem(): '1'인 유전자 중에 가장 의미 없는 (그것 하나를 뺐을 때 성능 저하가 가장 적은) 유전자를 찾아 '0'으로 한다.

add(): '0'인 유전자 중에 가장 의미 있는 (그것 하나를 더했을 때 성능 향상이 가장 큰) 유전자를 찾아 '1'로 한다.

REM(k): rem() 연산을 k번 반복한다.

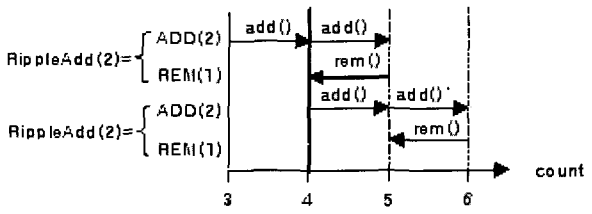
ADD(k): add() 연산을 k번 반복한다.

RippleRem(r) ≡ { REM(r); ADD(r-1); }, r≥1

RippleAdd(r) ≡ { ADD(r); REM(r-1); }, r≥1

rem 연산은 현재 선택된 앙상블에서 가장 의미 없는 분류기를 하나 제거하는 역할을 한다. 반대로 add 연산은 선택되지 않은 분류기들의 집합에서 가장 의미 있는 분류기를 선택하여 앙상블에 추가해주는 역할을 한다. count는 현재 앙상블에 선택된 분류기의 수를 의미한다. 원하는 분류기 앙상블의 크기가 d 이므로 |d-count| 만큼의 반복을 통해 분류기를 추가 또는 제거해준다. r은 탐색의 깊이를 조절하는 인자로 지역 탐색 성능 향상의 강도를 조절한다. r은 rem과 add 연산의 실행 횟수에 직접적인 영향을 주므로 r이 클수록 넓은 영역을 탐색하고 그에 따른 시간 비용도 증가한다.

(그림 3)은 r=2, count=3 이고 d=5 일 때의 지역 탐색 연산의 동작 예이다. 현재 선택된 분류기는 3이므로 선택되지 않은 분류기 풀에서 인식률 향상에 의미있는 분류기 2개를 선택해야 한다. RippleAdd(2) 연산은 탐색 깊이를 2로 하여



(그림 3) 지역 탐색 연산 동작 예

add() 연산을 두 번 하고, rem() 연산을 한 번 하게 된다. 즉, 분류기가 두 개 추가된 후, 한 개를 삭제 과정을 거쳐 count=4가 된다. for 반복문에 의해 RippleAdd(2)을 2번 실행하게 되고 count=5가 된다.

지역 탐색 연산은 r이 크면 클수록 더 많은 계산 시간을 필요로 하기 때문에 실행 시간을 고려해야 한다. 한편, 지역 탐색의 미세 조정 능력이 우수할수록 유전 알고리즘은 빨리 수렴하게 된다. 따라서 효율적인 혼합 유전 알고리즘을 설계하는데 가장 중요한 관점은 실행되는 연산의 수를 최소화하는 동시에 가능한 한 빠른 지역 탐색 연산을 설계하는 것이다.

2.2.2 조합 탐색 연산 (CSO: Combinational Search Operations)

앙상블은 한 개 이상의 인식기들이 결합하여 분류기의 인식 결과를 결정하므로 각 인식기가 독립적이지 않다. 이러한 이유로 분류기 풀을 생성할 때에도 다양성을 고려한다. 제안하는 또 하나의 지역 탐색 연산은 분류기 앙상블에 추가되거나 삭제되는 분류기를 결정할 때에 분류기들 사이의 상관관계를 고려한다. 순차 탐색은 한 개의 분류기가 추가 또는 삭제 될 때 다음에 들어올 분류기와의 관계를 고려하지 않고 현재 가장 좋은 인식률을 기준으로 탐색이 이루어진다. 반면 조합 탐색 연산은 한 개 이상의 분류기로 이루진 부집합의 추가 나 삭제 시 변화하는 인식률을 기준으로 탐색한다. 조합 탐색 연산은 |*d*-count| 크기의 분류기 부집합을 만들어 그들 중 우수한 부집합을 찾아 이를 추가하거나 삭제한다. 조합 탐색 연산은 아래와 같다.

```
combinational_local_search(offspring) {
  count = offspring에서 '1'인 유전자의 수;
  switch {
    case count=d: cadd(k); crem(k);  // k 는 작은 수
    case count<d: cadd(d-count);
    case count>d: crem(count-d);
  }
}
```

crem(k): 집합 X(offspring)로부터 크기가 k인 모든 부집합을 만들고, 이들 부집합으로부터 임의로 m개를 선택한다. 선택된 m개의 부집합 중에서 가장 의미 없는 부집합을 (그 부집합을 뺐을 때 성능 감소가 가장 작은) 찾아 그 부집합에 속한 유전자들을 '0'으로 바꾼다.

cadd(k): 집합 Y(offspring)로부터 크기가 k인 모든 부집합을 만들고, 이들 부집합으로부터 임의로 m개를 선택한다. 선택된 m개의 부집합 중에서 가장 의미 있는 부집합을 (그 부집합을 더했을 때 성능 증가가 가장 큰) 찾아 그 부집합에 속한 유전자들을 '1'로 바꾼다.

위에서 X(offspring)와 Y(offspring)는 각각 '1'값을 가진 유전자의 집합과 '0'값을 가진 유전자 집합을 의미한다. 위 연산에서 모든 부집합을 대상으로 가장 좋은 부집합을 찾는 대신, m개의 부집합을 임의로 선택한 후 그 중에서 가장 좋은 부집합을 찾는다. m의 값은 연산의 계산 시간과 관련되어 식2에 의해 조정된다. 부집합의 전체 개수는 $_{|X|}C_k$ 이므로 매우 큰 탐색 공간을 가질 수 있다. 따라서 그들 일부에서 가장 좋은 것을 선택할 수 있도록 하였다. 탐색 비율 q를 아래와 같이 정의한다. 식 2의 q가 1.0일 때는 모든 부집합을 대상으로 탐색하는 경우이다.

$$q = m/_{|X|}C_k \qquad \text{(식2)}$$

3. 실험 및 결과 분석

인식기로는 k-NN 분류기를 사용하였으며, 서로 다른 특징 부집합(feature subset)을 가진 *N*개의 k-NN 분류기로 분류기 풀을 구성하였다. k-NN 분류기의 거리 함수는 Euclidean Distance를 사용하였다. 특징 부집합은 전체 특징 집합에서 임의 수로 특징들을 선택하여 만들었다. HGA는 (그림 2)의 알고리즘에서 HYBRID 변수를 TRUE로 설정하여 지역 탐색 연산이 수행되도록 한 것 이외에 모든 조건은 SGA와 같게 하였다. 공정한 평가를 위해 SGA와 HGA의 수행 시간을 같게 하였다.

<표 1>은 실험을 위해 사용한 데이터 집합을 보여준다. 앞의 네 개는 UCI repository[20]에서 선택한 데이터 집합이고, 마지막 하나는 CENPARMI 필기 숫자 집합이다.

〈표 1〉 실험에 사용한 데이터 집합

	샘플 개수	훈련 개수	검증 개수	성능평가	특징 개수	부류 개수
WDBC	569	569	0		30	2
IoNosphere	351	351	0		34	2
Sonar	208	208	0	Leave-one-out	60	2
Letter	20000	15000	5000		16	26
CENPARMI	6000	4000	2000		512	10

〈표 2〉 분류기 풀에 있는 1-NN 분류기의 성능

데이터집합	분류기 풀			
	특징 부집합의 크기	최소	최대	평균
Letter	8	85.01%	92.29%	87.36%
WDBC	15	90.16%	93.67%	91.51%
IoNosphere	17	84.33%	90.59%	87.94%
Sonar	30	85.09%	87.50%	86.18%
CENPARMI	256	92.90%	95.10%	94.02%

<표 2>는 1-NN 분류기 풀의 성능을 보여준다. 분류기 풀의 크기(N)는 50으로 하였고, 특징 부집합의 크기는 원래 특징 집합 크기의 1/2로 하였다. <표 2>에서 분류기 풀에 속한 분류기의 최대, 최소 그리고 평균 인식률을 보여 준다.

<표 3>은 각 데이터 집합에 대한 SGA와 HGA의 성능 비교이다. HGA의 지역 탐색 연산은 2.2절에서 설명한 순차 탐색 연산(SSO)과 조합 탐색 연산(CSO)에 대하여 각각 실험하였다. SSO에서 r은 2로 하였고 CSO의 q는 0.5, k는 3으로 하였다. 각 분류기 풀에 대해 앙상블의 크기가 4, 8, 12, 16, 20그리고 24 일 때로 나누어 성능을 측정하였고, 보다 객관적인 성능 비교를 위해 인식률은 다섯 번의 독립적인 실행의 평균값으로 하였다. 괄호 속은 다섯 번의 실행 중 최대 인식률이다.

<표 2>와 <표 3>의 비교를 통해 분류기 앙상블이 성능 향상에 효과적임을 알 수 있다. 예를 들어, Letter 데이터 집합의 경우 평균 87.36%의 분류기들을 결합하여 앙상블 인식률 최대 98%까지 얻을 수 있었다. WDBC의 경우는 평균 91.51%의 분류기들을 이용해 앙상블 인식률 최대 96.14%, IoNosphere의 경우 평균 87.94%의 분류기들을 결합해서 앙상블 인식률 최대 95.16%를 얻었다. Sonar의 경우 평균 86.18%의 분류기들을 이용해 앙상블 인식률 최대 92.79%, CENPARMI 데이터의 경우는 평균 94.02%의 분류기들을 결합해 앙상블 인식률 최대 97.4%를 얻었다. 실험에 사용한 결합 방법은 단순한 투표 방법을 사용했음에도 분류기 앙상블은 인식률의 향상에 효과적임을 알 수 있다.

<표 3>과 (그림 4)는 앙상블의 크기가 작을 때 몇 경우

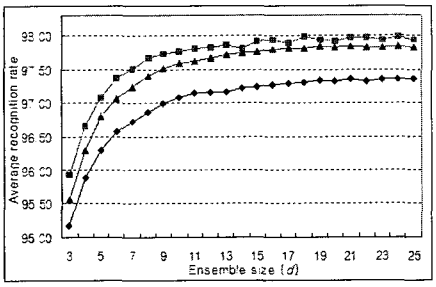
〈표 3〉 SGA와 HGA의 인식률 비교

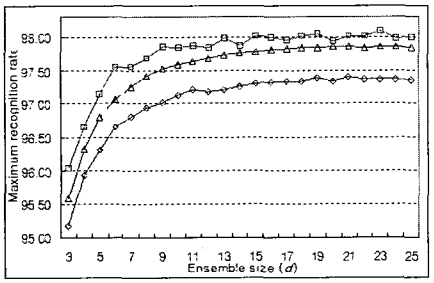
(SGA: 단순 유전 알고리즘, HGA: 혼합 유전 알고리즘, SSO: 순차 탐색 연산, CSO: 조합 탐색 연산)

데이터집합	분류기 풀 크기(N)	d	SGA	HGA	
				SSO	CSO
Letter	50	4	95.89(95.93)	96.30(96.33)	96.65(96.66)
		8	96.87(96.95)	97.40(97.42)	97.66(97.68)
		12	97.16(97.19)	97.67(97.68)	97.83(97.84)
		16	97.28(97.32)	97.80(97.81)	97.94(98.00)
		20	97.32(97.34)	97.83(97.85)	97.92(97.94)
		24	97.35(97.37)	97.84(97.86)	97.98(97.98)
WDBC	50	4	95.11(95.43)	95.36(95.43)	95.15(95.43)
		8	95.61(95.78)	95.92(95.96)	95.99(96.14)
		12	95.47(95.61)	95.89(95.96)	95.96(95.96)
		16	95.43(95.61)	95.68(95.78)	95.89(95.96)
		20	95.39(95.61)	95.61(95.78)	95.64(95.78)
		24	95.33(95.43)	95.50(95.61)	95.64(95.78)
IoNosphere	50	4	93.16(93.45)	93.45(93.45)	93.45(93.45)
		8	93.96(94.02)	94.53(94.59)	94.59(94.59)
		12	94.36(94.59)	94.70(94.87)	95.16(95.16)
		16	94.36(94.59)	94.76(94.87)	94.93(95.16)
		20	94.25(94.59)	94.70(94.87)	94.87(94.87)
		24	94.19(94.59)	94.53(94.87)	94.81(94.87)
Sonar	50	4	91.06(91.83)	91.83(91.83)	91.83(91.83)
		8	91.35(91.83)	92.34(92.79)	92.59(92.79)
		12	91.73(92.31)	92.30(92.79)	92.69(92.79)
		16	91.15(91.83)	92.40(92.79)	92.69(92.79)
		20	90.96(91.35)	91.83(92.31)	92.31(92.31)
		24	90.67(90.87)	91.54(91.83)	91.83(91.83)
CENPARMI	50	4	96.69(96.70)	96.90(96.90)	97.00(97.00)
		8	97.02(97.10)	97.22(97.30)	97.40(97.40)
		12	97.06(97.10)	97.30(97.30)	97.40(97.40)
		16	97.06(97.10)	97.26(97.30)	97.38(97.40)
		20	97.06(97.10)	97.22(97.30)	97.36(97.40)
		24	96.98(97.10)	97.18(97.20)	97.30(97.30)

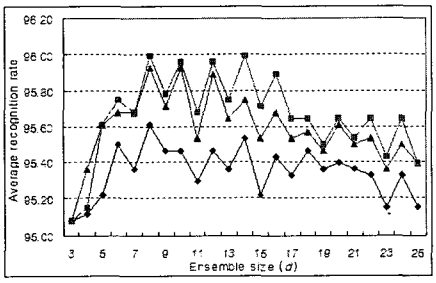
만 (예를 들어, d=4일 때 WDBC, IoNosphere, Sonar 데이터 집합) 제외하고는 HGA가 SGA보다 우수함을 보이고 있다. HGA와 SGA의 성능 차이는 평균값에서는 0.25~1.54% 이며, 최대값에서는 0.00~0.98% 정도이다. SSO와 CSO의 성능을 비교해 보면 대체로 CSO가 조금 더 우수함을 알 수 있다. 실험 결과를 분석해 본 결과 CSO 방법을 사용한 HGA에서 최대값을 얻을 수 있었다.

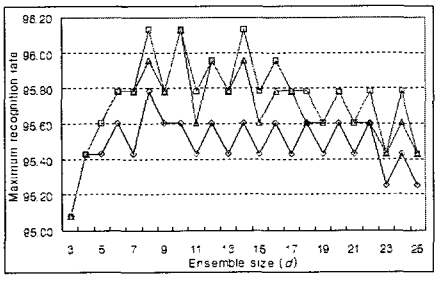
또한 (그림 4)는 분류기 앙상블의 최적의 크기를 보여준다. Letter 데이터 집합의 경우 앙상블 크기 20 정도 에서 최적을 보이고 다른 데이터 집합들은 앙상블 크기 10 정도에서 가장 높은 인식률을 보인다. 이는 d가 크다고 반드시 좋은 성능을 보이지는 않는다는 사실을 알려준다. 이 사실은 논문[2]의 주장과 일치한다고 할 수 있다.



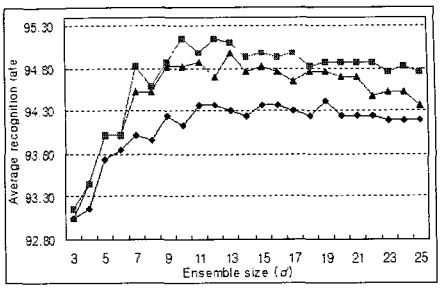


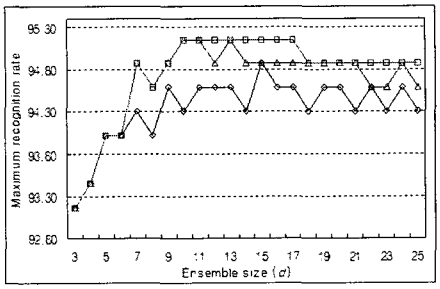
(a) Letter



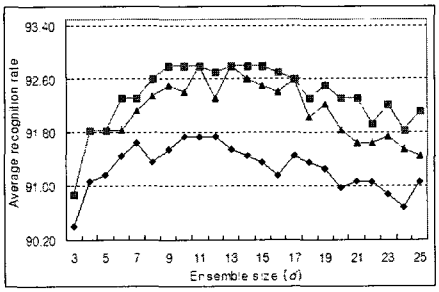


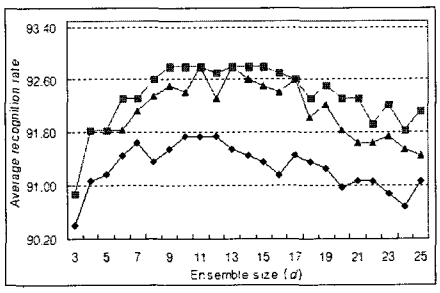
(b) Letter



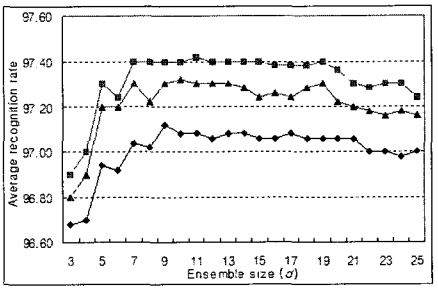


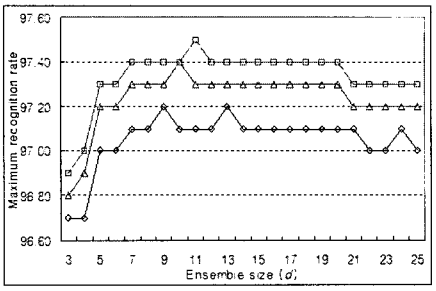
(c) IoNosphere



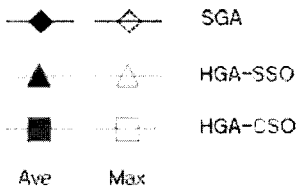


(d) Sonar





(e) CENPARMI



(그림 4) 인식 성능 비교

<표 4>는 각 알고리즘의 안정성을 관찰하기 위해 다섯 번의 독립적인 실험으로 얻은 인식률의 표준편차를 계산한 것이다. 값이 작을수록 알고리즘이 안정적으로 동작함을 의미한다. <표 4>는 HGA가 SGA보다 안정적이고 CSO가 SSO 보다 더 안정적임을 보여준다

〈표 4〉 표준편차×1000

데이터집합	분류기 풀 크기(N)	d	SGA	HGA	
				SSO	CSO
Letter	50	4	0.35	0.34	0.11
		8	0.48	0.26	0.22
		12	0.34	0.26	0.11
		16	0.30	0.10	0.33
		20	0.20	0.23	0.09
		24	0.19	0.12	0.05
WDBC	50	4	1.93	1.57	1.57
		8	2.15	0.79	0.79
		12	1.93	0.96	0
		16	1.24	0.96	0.96
		20	1.47	1.24	0.79
		24	0.96	0.96	0.79
IoNosphere	50	4	2.85	0	0
		8	1.27	1.27	0
		12	1.27	1.56	0
		16	1.27	1.56	1.27
		20	2.38	1.56	0
		24	2.55	3.12	1.27
Sonar	50	4	4.30	0	0
		8	5.89	2.83	2.64
		12	4.02	3.40	2.15
		16	4.30	2.15	2.15
		20	4.02	3.40	0
		24	2.63	2.63	0
CENPARMI	50	4	0	0	0
		8	0.45	0.45	0
		12	0.55	0	0
		16	0.55	0.55	0.45
		20	0.55	0.45	0.55
		24	0.84	0.45	0

4. 결 론

이 논문은 최적의 분류기 앙상블 선택을 위한 혼합 유전 알고리즘을 제안하였다. HGA와 SGA의 비교 실험을 통해 분류기 앙상블 선택에서 SGA에 비해 HGA가 해를 탐색하는 능력이 훨씬 뛰어나고 안정성도 높음을 입증하였다. 또한 HGA의 지역 탐색 연산으로 두 가지 방법을 제안하였으며, 실험을 통해 분류기들의 상관관계를 고려한 CSO 방법이 더 좋은 결과를 보였다. 향후 연구로 분류기 풀 구성에서 N의 크기 결정, 우수한 해를 찾아 분류기 풀을 구성하는 연구 등이 있다. 우수한 해를 찾기 위해서는 기존에 개발되어 있는 특징 선택 방법을 도입해 사용할 수 있다. 또한 최적의 d 값을 자동으로 결정하는 방법의 개발도 필요하다.

참 고 문 헌

[1] P.M. Granitto, P.F. Verdes, and H.A. Ceccatto, "Neural network ensembles: evaluation of aggregation algorithms," *Artificial Intelligence*, Vol.163, No.2, pp.139-162, 2005.

[2] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, Vol.137, pp.239-263, 2002.

[3] Chanho Park and Sung-Bae Cho, "EVolutionary ensemble classifier for lymphoma and colon cancer classification," *Proc. of IEEE International Conf. on EVolutionary Computation*, Vol.4, pp.2378-2385, 2003.

[4] N. Ueda, "Optimal linear combination of neural networks for improving classification performance," *IEEE Tr. Pattern Analysis and Machine Intelligence*, Vol.22, No.2, pp.207-215, 2000.

[5] L. Breiman, "Bagging predictors," *Machine Learning*, Vol.24, No.2, pp.123-140, 1996.

[6] Robert Bryll, Ricardo Gutierrez-Osuna, and Francis Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, Vol.36, No.6, pp.1291-1302, 2003.

[7] J. R. Quinlan, "Bagging, boosting, and C4.5," *Proc. AAAI-96*, pp.725-730, 1996.

[8] Tin Kam Ho, "Multiple classifier combination: lessons and next steps," in *Hybrid Methods in Pattern Recognition*, (Ed. by H. Bubke & A. Kandel), World Scientific, 2002.

[9] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Tr. Pattern Analysis and Machine Intelligence*, Vol.20, No.8, pp.832-844, 1998.

[10] G. Tremblay, R. Sabourin, and P. Maupin, "Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm," In *Proc. of 17th ICPR*, Vol.1, pp.208-211, 2004.

[11] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Tr. Pattern Analysis and Machine Intelligence*, Vol.20, No.3, pp.226-239, 1998.

[12] Antanas Verikas, Arunas Lipnickas, Kerstin Malmqvist, Marija Bacauskiene, and Adas Gelzinis, "Soft combination of neural classifiers: a comparative study," *Pattern Recognition Letters*, Vol.20, No.4, pp.429-444, 1999.

[13] S. Wesolkowski and K. Hassanein, "A comparative study of combination schemes for an ensemble of digit recognition neural networks," *Proc. of IEEE International Conf. on Computational Cybernetics and Simulation*. Vol.4, pp.3534-3539, 1997.

[14] G. Fumera and F. Roli, "A theoretical and experimental analysis of linear combiners for multiple classifier systems," *IEEE Tr. Pattern Analysis and Machine Intelligence*, Vol.27, No.6, pp.942-956, 2005.

[15] Hee-Joong Kang and David Doermann, "Selection of classifiers for the construction of multiple classifier systems," *Proc. of International Conf. on Document Analysis and Recognition*, Vol.2, pp.1194-1198, 2005.

[16] Il-Seok Oh, Jin-Seon Lee, and Byung-Ro Moon, "Hybrid genetic algorithms for feature selection," *IEEE Tr. Pattern Analysis and Machine Intelligence*, Vol.26, No.11, pp.1424-1437, 2004.

[17] P. Jog, J. Suh, and D. Gucht, "The effect of population size, heuristic crossover and local improvement on a genetic algorithm for the traveling salesman problem," *Proc. of International Conference on Genetic Algorithms*, pp.110-115, 1989.

[18] T.N. Bui and B.R. Moon, "Genetic algorithm and graph partitioning," *IEEE Tr. Computers*, Vol.45, No.7, pp.841-855, 1996.

[19] X. Zheng, B.A. Julstrom, and W. Cheng, "Design of vector quantization codebooks using a genetic algorithm," *Proc. of IEEE International Conf. on EVolutionary Computation*, pp.525-529, 1997.

[20] http://www.ics.uci.edu/~mlearn/databases/

[21] Yifeng Zhang and Siddhartha Bhattacharyya, "Genetic programming in classifying large-scale data: an ensemble method," *Information Sciences*, Vol.163, pp.85-101, 2004.

김 영 원

e-mail : everywkim@etri.re.kr

2001년 전북대학교 컴퓨터과학과(학사)

2003 전북대학교 컴퓨터정보학과(석사)

2006.8 전북대학교 컴퓨터통계정보학과 (박사)

2006년 9월~2007년 6월 전북대학교 BK21 사업단 Post-Doc

2007년 7월~현재 한국전자통신연구원 우정기술연구센터 연구원

관심분야 : 워터마킹, 문자인식, 유전 알고리즘, 컴퓨터비전

오 일 석

e-mail : isoh@chonbuk.ac.kr

1984년 서울대학교 컴퓨터공학과(학사)

1992년 KAIST 전산학과(석사, 박사)

1992년 9월~현재 전북대학교 전자정보공학부 교수

2005년 1월~2006년 12월 한국정보과학회 컴퓨터비젼 및 패턴인식 연구회 운영위원장

2006년 9월~현재 한국콘텐츠학회 논문지 편집위원장

2004년 1월~2004년 12월 한국정보과학회 SA 논문지 편집위원장

관심분야 : 문서영상 처리, 패턴인식, 유전알고리즘의 패턴인식 응용