

## Comparison of Various Criteria for Designing ECOC<sup>1)</sup>

Kyeong Ha Seok<sup>2)</sup> · Seung Chul Lee<sup>3)</sup> · Gab Dong Jeon<sup>4)</sup>

### Abstract

Error Correcting Output Coding(ECOC) is used to solve multi-class problem. It is known that it improves the classification accuracy. In this paper, we compared various criteria to design code matrix while encoding. In addition, we propose an ensemble which uses the ability of each classifier while decoding. We investigate the justification of the proposed method through real data and synthetic data.

**Keywords** : Codematrix, Error correcting output coding(ECOC), Multi-class problem

### 1. 서 론

클래스가 3개 이상인 다중분류문제(multi-class problem)를 해결할 수 있는 분류기(classifier)의 설계가 최근에 많은 관심을 가지는 연구과제 중의 하나이다. 이를 해결하는 방법 중 하나는 Diettrich와 Bakiri (1991, 1995)에서 제안한 ECOC방법인데 이는 하나의 다중분류문제를 여러 개의 이진분류부문제(binary classification sub-problem)로 분할하여 이진 분류문제를 해결한 후, 그 결과를 다시 합치는 방법이다. ECOC는 다음과 같은 타당한 이유 때문에 많은 관심을 받고 있다. 첫 번째는 정확성과 효과를 지닌 이진분류기라 할지라도 다중분류기로 확장되는 것이 어렵다는 것이고 두 번째는 이진분류기에서의 작업이 상대적으로 용이하다는 것이다. 그리고 세 번째는 목표값(target)을 이진 값으로 바꾼 자료를 여러 번 반복 훈련함으로써 오류율(error rate)을 줄일 수 있다는 것이다.

ECOC는 패턴인식 분야에서부터 개발되어 최근에는 문자분류 등 많은 분야에서 성

---

1) This work supported by Inje Research Grant 2004.

2) Department of Data Science, Inje University, Kyungnam, 621-749, Korea, 통계정보연구소.  
E-mail : statskh@paran.com

3) Graduate student, Department of Data Science, Inje University, Kyungnam, 621-749, Korea

4) Graduate student, Department of Data Science, Inje University, Kyungnam, 621-749, Korea

공적으로 사용되고 있다(Kittler 등(2001), Aha와 Bankert(1997), Windeatt와 Ghaderi(2003), Berger(1999)). Diettrich와 Bakiri(1991, 1995)는 ECOC에 대해 소개하고, 이를 설계하는 기본적인 방법들을 비교하고 있다. 그리고 여러 가지 실험을 통해서 ECOC가 다중분류문제에 잘 적용될 수 있음을 보여주었다. Windeatt와 Ghaderi(2003)은 효율적인 ECOC의 설계방법을 제시하였을 뿐 아니라, 다양한 decoding방법에 대해 소개하고 있다. Kuncheva(2005)는 코드행렬(codematrix) 설계에서 다양한 축도의 활용방안을 제시하고 실험을 통하여 제시된 방법을 비교하였다.

본 논문은 코드행렬 설계에 사용되는 다양한 축도에 대해 소개하고, 모의실험과 실제자료(real data)를 이용한 비교를 통해 명확한 판단기준을 정립하는데 그 목적을 두고 있다. 그리고 각 분류기의 성능을 decoding과정의 결합기에 반영함으로써, 다중분류 작업의 정확성을 높이는 방법을 제안한다.

본 논문의 2장에서는 ECOC에 대해 소개하고, 3장에서는 ECOC 설계기준에 대해서 소개를 하고 분류기의 성능을 반영하는 결합기를 제시한다. 그리고 4장에서는 모의자료와 실제자료를 통한 실험의 결과를 보여준다.

## 2. ECOC

ECOC는 다중분류 문제는 여러 개의 이진분류부문제로 변환하는 encoding 단계와 이진분류부문제의 결과를 결합하여 원래의 클래스라벨(class label)로 환원하는 decoding단계로 나누어진다.

$\Omega = \{1, 2, \dots, c\}$ 를 클래스 라벨(class label)의 집합이라 하자. 각 이진분류 부문제는 각각의 이진분류기를 통해서 해결되는데, 각 분류기의 목표값은 이진형 값, 즉  $\{-1, 1\}$ 의 값을 갖는다. 예를 들어,  $\Omega = \{1, 2, 3, 4\}$ 인  $c=4$ 의 다중분류문제를 고려해보자. 그러면 목표값  $y$ 는  $\{1, 2, 3, 4\}$  중 하나의 값을 갖게 된다. 이 문제를 <표 1>과 같은 행렬을 이용하여 이진분류문제로 변환시킨다고 하자. 이러한 행렬을 코드행렬이라고 한다.

<표 1>  $c=4, L=7$ 인 코드행렬

	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$
1	-1	-1	-1	1	-1	1	1
2	-1	-1	1	-1	-1	-1	-1
3	-1	1	-1	-1	1	-1	1
4	1	-1	-1	-1	1	1	-1

첫 번째 열은 이진분류기  $B_1$ 의 목표값이 되는데 편의상 분류기  $B_1$ 이라고 하자. 이는 원래의 클래스라벨이  $\{1, 2, 3\}$ 인 것은  $\{-1\}$ 로,  $\{4\}$ 인 것은  $\{1\}$ 로 하는 이진분류문제를 해결하는 것이다. 이러한 방법으로 7개의 분류기로 이진문제를 해결한

후 결합기(ensemble)를 이용하여 원래의 라벨을 추정한다. 코드행렬의 각 행을 코드워드(codeword)라고 하며,  $C_i, i = 1, 2, \dots, c$ 로 표기한다. 이는 원래의 클래스라벨이 각 분류기에서 어떻게 변환되는지를 보여준다.

시험자료(test data)  $x$ 가 입력되면,  $L$ 개의 각 분류기에서 출력값이 생성된다. 즉,  $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_L)$ 가 계산되는데, 여기서  $\hat{y}_i$ 는  $i$ 번째 분류기에 의한 실수 혹은 이진 출력값을 의미한다. 출력값  $\hat{y}$ 는 각 코드워드간의 거리를 계산하여, 가장 작은 값에 해당하는 코드워드의 클래스로  $x$ 를 분류한다. 이를 decoding단계라 한다. 거리를 나타내는 측도로써 여러 가지가 사용될 수 있지만 (Allwein 등(2000), Kuncheva와 Whitaker (2003)) 해밍거리(Hamming Distance)가 가장 보편적인데 이는

$$H = \sum_{i=1}^n |a_i - b_i|/2 \text{으로 정의된다. 여기서 } a = (a_1, a_2, \dots, a_n) \text{와 } b = (b_1, b_2, \dots, b_n) \text{는 } -1 \text{과 } 1 \text{을 원소로 하는 임의의 두 벡터이다.}$$

대표적인 코드행렬은 OPC(One Per Class)와 APC(All Pairwise Comparison), 그리고 EC(Exhaustive Coding)이 있다. OPC는 각 분류기가 하나의 클래스만  $\{1\}$ 로 나머지는  $\{-1\}$ 로 하여 문제를 해결하는 방법으로서,  $c=4$ 일 때 다음의 <표 2>와 같은 코드행렬을 사용한다.

<표 2>  $c=4$ 인 OPC 코드행렬

	$B_1$	$B_2$	$B_3$	$B_4$
1	1	-1	-1	-1
2	-1	1	-1	-1
3	-1	-1	1	-1
4	-1	-1	-1	1

APC는 가능한 모든 클래스의 쌍을 이용하는 방법으로서, <표 3>(  $c=3$ )과 같이 사용하지 않는 클래스를  $\{0\}$ 으로 표현한다(Allwein 등(2000)).

<표 3>  $c=3, L=3$ 인 APC 코드행렬

	$B_1$	$B_2$	$B_3$
1	1	1	0
2	-1	0	1
3	0	-1	-1

EC는  $c$ 클래스에 대하여 모든 가능한  $2^{(c-1)} - 1$  경우의 부분제로 나타내는 방법

으로 예를 들어  $c=4$ 인 EC 코드행렬은 다음의 <표 4>와 같다.

<표 4>  $c=4, L=7$ 인 EC 코드행렬

	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$
1	1	1	1	1	1	1	1
2	-1	-1	-1	-1	1	1	1
3	-1	-1	1	1	-1	-1	1
4	-1	1	-1	1	-1	1	-1

$c=3$ 일 때 EC와 OPC가 동일한 결과를 가진다. 이는  $c=3$ 인 경우에는 ECOC로부터 얻을 수 있는 효과가 그리 크지 않다는 것을 의미한다.  $3 \leq c \leq 7$ 인 경우에는 EC를 이용한 방법이 주로 사용되고,  $8 \leq c \leq 11$ 의 경우 또한 EC를 이용한 선택방법을 권하고 있다.  $c > 11$ 인 경우에 대해서는 랜덤코드 생성이 효과적인데 코드워드의 랜덤 결정법의 효과에 대해서는 많은 연구결과들이 이를 증명하고 있다 (Diettrich와 Bakiri (1995), Windeatt 와 Ghaderi (2003), Schapire(1997)).

### 3. 코드행렬 설계

코드워드  $C_i$ 와  $C_j$ 의 해밍거리는

$$HC_{ij} = \sum_{k=1}^L |C(i, k) - C(j, k)| / 2 \quad , \quad i, j = 1, 2, \dots, L$$

로 정의를 할 수 있는데, 이 값이 커지도록 코드행렬을 설계를 해야 한다. 그렇게 함으로써 이진분류기에서 발생하는 오류를 극복할 수 있다.  $HC_{ij}$ 의 최소값이  $d$ 일 때  $[(d-1)/2]$ 개의 오류를 극복할 수 있다(Diettrich와 Bakiri (1995)). 여기에서  $[x]$ 는  $x$ 를 초과하지 않는 최대정수이다. 이 값을 오차수정력(error correcting power)이라 하며, 오류를 범한 분류기의 개수가 오차수정력 이하이면 결합기가 정분류를 할 수 있으므로  $d$ 를 크게 하도록 코드행렬을 설계해야 한다.

또한 각각의 분류기들이 독립적일 때 코드행렬은 더욱 의미가 있다. 그러므로 ECOC를 설계하는데  $B_i$ 와  $B_j$ 의 해밍거리는

$$HB_{ij} = \min_{i,j} \min \left\{ \sum_{k=1}^c |C(k, i) - C(k, j)|, \sum_{k=1}^c |2 - C(k, i) - C(k, j)| \right\}, \quad i, j = 1, 2, \dots, L$$

로 정의하는 것이 타당하다. 분류기의 상관관계를 낮추는 다른 방법에는 훈련용자료(training data)를 재표본추출(resampling)하는 Bagging(Breiman (1997))과 Boosting방

법(Freund와 Schipire(1997))이 있다.

이상에서 설명한 것처럼 코드행렬을 설계할 때 열간의 거리와 행간의 거리를 최대한으로 하는 것이 목표가 된다. 이를 위해서  $AHC$ 와  $AHB$ 를  $HC_{ij}$ 와  $HB_{ij}$ 의 평균이라고 하고  $LHC$ 와  $LHB$ 를  $HC_{ij}$ 와  $HB_{ij}$ 의 최소값이라고 하자. 그리고

$$L_\alpha = \alpha LHC + (1-\alpha)LHB, \quad A_\alpha = \alpha AHC + (1-\alpha)AHB, \quad 0 \leq \alpha \leq 1$$

라 하자.

거리의 가중평균을 나타내는  $A$ 와 최소거리의 가중평균을 나타내는  $L$ 중 어떤 것을 사용하는 것이 좋은지에 대해서 고려해 본다. 다음의 <표 5> CM1은 OPC와의 비교를 위해 임의로 만들어진 코드행렬이다(Kuncheva(2005)).

<표 5>  $c=5, L=5$ 인 CM1

	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$
1	1	-1	-1	-1	-1
2	1	1	1	1	-1
3	1	-1	1	1	1
4	-1	-1	-1	-1	-1
5	-1	1	-1	1	1

OPC는  $AHC = 0.4, AHB = 0.4, LHC = 2, LHB = 2$ 이고, CM1은  $AHC = 0.6, AHB = 0.32, LHC = 1, LHB = 2$ 의 값을 가진다.  $A_{1/2}$  값이 큰 CM1이 OPC 보다 더 정분류율이 높은 실험결과를 보였다(Kuncheva(2005)). 그러나 수행능력이 좋지 않은 것으로 알려진 OPC와의 비교결과를 일반적인 것으로 인정하기가 어렵기 때문에 본 논문에서는 여러 상황에서  $A$ 와  $L$ 기준 중 어떤 기준이 더 우수한지, 더 나아가 가중치를 서로 다르게 하는 것이 어떤 의미가 있는지를 살펴보고자 한다.

결합기(ensemble)는 출력값과 각 코드워드간의 거리를 어떻게 정의하느냐가 매우 중요하며 여러 가지가 고려 될 수 있지만 여기에서는 해밍거리를 사용한 것을 고려하기로 한다. 5개의 분류기를 통한 출력값이  $(1, 1, 1, 1, 1)^T$ 이라 할 때, 해밍거리가 1인  $C_1 = (-1, 1, 1, 1, 1)^T$  과  $C_2 = (1, -1, 1, 1, 1)^T$ 가 있다고 하자. 기존의 방법은 라벨 1과 2중 하나로 랜덤하게 분류하는 것이다. 그러나 만약에  $B_1$ 의 정분류율이 0.9이고 분류기  $B_2$ 의 정분류율이 0.6이라면, 출력값을 2로 분류해야 하는 것이 타당하다. 즉, 결합기는 각 분류기의 성능을 고려하는 것이 타당하며, 따라서 본 논문에서는 결합기를

$$\operatorname{argmin}_k \sum_{i=1}^L a_i |\widehat{y}_i - C(k, i)|$$

로 사용함을 제안한다. 여기에서  $a_i$ 는  $i$ 번째 분류기  $B_i$ 의 정분류율이다.

다음 장에서는 코드행렬설계를 위한 여러 가지 기준과 제안된 결합기의 수행능력을 실험을 통해 살펴본다.

#### 4. 실험결과

다음 실험의 목적은 첫 번째가  $A$ 와  $L$ 간 어떠한 차이가 있는지를 알아보는 것이고, 두 번째는 각 분류기의 성능을 고려한 제안된 결합기의 효과를 측정하기 위한 것이다. 이 실험을 위해 모의자료와 glass자료(ftp://ftp.ics.uci.edu/pub/machine-learning-databases)를 사용하였다. 모의자료를 이용한 실험에서는  $c$ 를 5, 6, 10으로 하였다. 먼저  $c=5$ 이고 각 분류기의 정분류율이 0.7인 경우의 모의실험에 대해서 간단히 설명하면 아래와 같다.

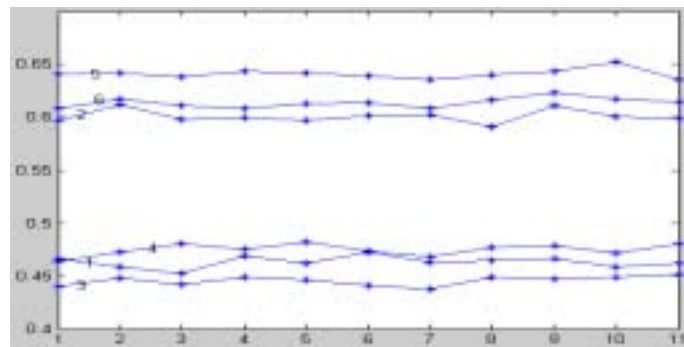
- (1) 클래스라벨을 랜덤하게 선택한다. 선택된 라벨을  $i$ 라 표기한다.
- (2)  $i$ 번째 코드워드의 각 원소값이 원래의 값을 가질 확률은 0.7이고, 값이 바뀔 확률은 0.3가 되는 선택을 한다.
- (3) 만약 한 개의 오류도 발생하지 않는다면 결합기는 클래스라벨  $i$ 를 출력한다. 많은 오류가 발생한다면, 출력값과  $C_i$ 가 서로 다른 값을 갖게 된다. 그러면 출력값과 모든 코드워드간의 거리를 고려하는 결합기가 다른 클래스라벨을 출력할 수 있다. 만약 출력값이 둘 이상의 코드워드와 동일한 거리일 경우에는 랜덤으로 배정한다.
- (4) 10,000번을 반복 시행하여, 정분류율을 계산한다.

본 논문에서는  $\alpha$ 값이 0, 0.3, 0.5, 0.7, 1의 값을 가질 때의 성능변화에 대해 알아보았다. 본 실험에서  $c=5, 6, 10$ 에서 적당한 분류기 개수  $L$ 개를 사용하였다. 그리고 1,000개의  $c \times L$  랜덤이진행렬 중 기준값이 가장 큰 코드행렬을 각 기준에 적당한 코드행렬로 선택하고 편의상 다음과 같이 번호를 부여하였고(번호 =  $c \times L$ , 즉  $1=5 \times 5, 2=5 \times 10, 3=6 \times 6, 4=6 \times 7, 5=6 \times 15, 6=10 \times 20$ ), 이 번호는 <그림 1>과 <그림 2>에 표시되어 있다.

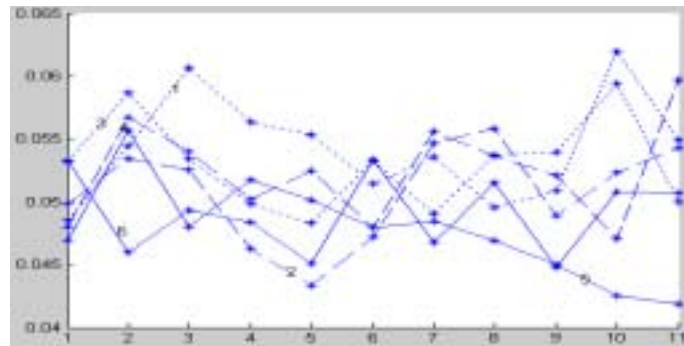
이 실험의 결과는 <그림 1>과 <그림 2>와 같다. 이 그림에서 각 분류기의 정분류율은 0.7이며,  $x$ 축의 번호는  $A$ 와  $L$ 각각의  $\alpha$ 값이 0, 0.3, 0.5, 0.7, 1인 경우와  $AHC + AHB + LHC + LHB$ 을 기준으로 했을 때의 결과이다. 예를 들어  $x$ 축 번호 1, 7은 각 기준  $A_0, L_{0.3}$ 을 나타낸다. 그림에서 알 수 있듯이  $A, L$  그리고,  $\alpha$ 의 선택은 결과에 큰 영향을 주지 않는다. 이는  $\alpha=0.5$ 인  $A_{0.5}$ 의 기준이 우수하

다는 Kuncheva(2005)의 연구결과가 타당하지 않음을 의미한다.

<그림 1>에서는 특이한 경향을 찾을 수 없다. 그러나 동일한  $c$ 에 대해서  $L$ 의 값이 클 때 정분류율이 더 높은 것으로 나온다. <그림 2>에서  $c=6$ 일 때  $L=6, 7, 15$ 에 따른 표준편차를 보면,  $L$ 이 클수록 표준편차가 작아짐을 알 수 있다. 그리고  $L_0$ 와  $A_0$ 기준(1번과 6번)이 가장 안정적으로 보이고,  $A_{0.5}$ 와  $L_1$ (3번과 10번)기준은 변동이 가장 심한 것으로 나타났다.  $x=3$ 과 8인  $A_{0.5}$ 와  $L_{0.5}$ 를 비교했을 때, 정분류율의 평균은 비슷하지만,  $L_{0.5}$ 가 조금 더 안정적이라는 것을 알 수 있다.



<그림 1> 정분류율 0.7인 분류기의 정분류율 평균

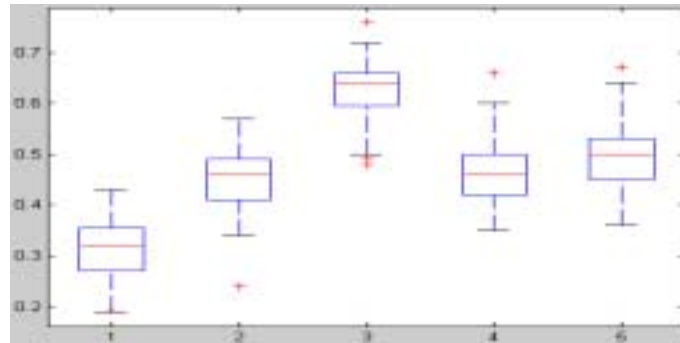


<그림 2> 정분류율 0.7인 분류기의 정분류율 표준편차

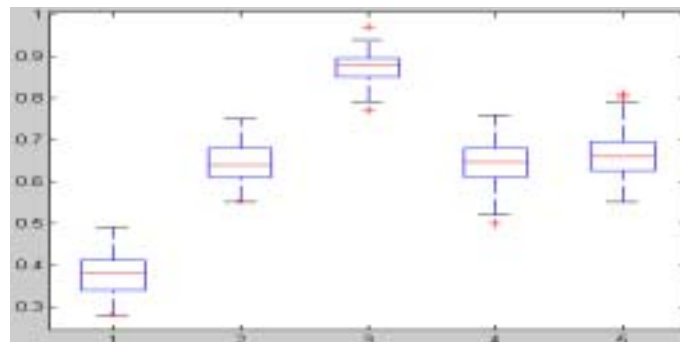
본 논문에서 제안된 결합기의 수행결과를 알아보기 위한 실험의 결과는 다음의 <그림 3>과 <그림 4>에 상자그림으로 표현하였다. <그림 3>은  $L=5, A_{0.5}$  기준인 경우를 나타낸 것이다.  $x$ 축의 번호는 분류기의 성능에 따라 나누어지는데, 1, 2, 3번은 모든 분류기의 정분류율이 각각 0.6, 0.7, 0.8인 경우이고, 4는 정분류율이 0.6에서 0.8까지 선형적으로 증가하면서 각 분류기의 정분류율이 서로 다른 것, 그리고 마지막 5번은 4번의 조건과 동일하나 제안된 결합기의 결과이다.

실험의 결과를 상자그림으로 나타내었는데 <그림 3>에서 제안된 방법의 정분류율이 2, 4번째보다 조금 크다는 것을 알 수 있다. 그리고 각각의 방법에 대한 정분류율

은 (0.3166, 0.4519, 0.6266, 0.4659, 0.4957)이고 표준편차는 (0.0522, 0.0607, 0.0552, 0.0593, 0.0623)이다. 제안된 방법의 정분류율이 조금 증가하지만 표준편차도 조금 증가하는 경향이 있다. <그림 4>는  $(c, L) = (6, 15)$ 일 때  $A_{0.7}$ 기준의 결과를 보인 것인데 <그림 3>과 비슷한 결과를 보이며, 각 방법의 정분류율은 (0.3775, 0.6436, 0.8757, 0.6477, 0.6660)이고, 표준편차는 (0.0435, 0.0450, 0.0340, 0.0506, 0.0554)이다.



<그림 3>  $(c, L) = (5, 5)$ 일 때,  $A_{0.5}$ 기준의 결과



<그림 4>  $(c, L) = (6, 15)$ 일 때,  $A_{0.7}$ 기준의 결과

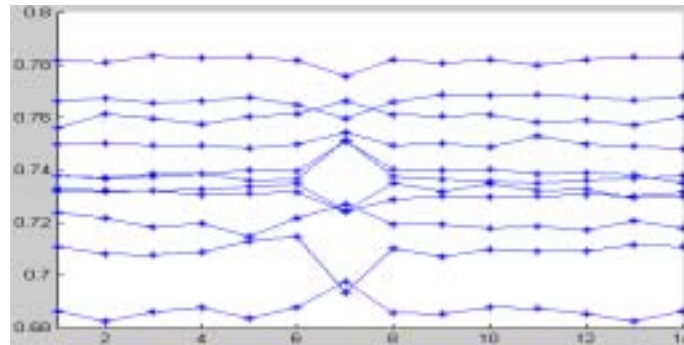
실제자료에 적용시켰을 때의 결과를 알아보았다. 크기가 214, 클래스수가  $c=6$ 인 glass자료를 이용하였으며, 분류기로는 LS-SVM(Suykens, 2002)을 사용하였다. LS-SVM를 사용할 때 필요한 커널모수  $\sigma$ 와 정칙화(regularization)모수  $\mu$ 는 사전 실험을 통해 적당한 값으로 선택된 것을 이용하였다.

좀 더 자세한 비교를 위해  $\alpha$ 가 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1일 때,  $A_\alpha$ 와  $L_\alpha$ 의 기준을 사용하여 100번의 반복실험을 하였다. <그림 5>와 <그림 6>은  $L = [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20]$ ,  $\sigma=0.5$ 일 때의 정분류율 평균과 표준편차이다.

<그림 5>는  $x$ 축 방향으로 큰 차이를 보이지 않고, 단지 7번째  $\alpha=1$ 인  $A_1$ 기

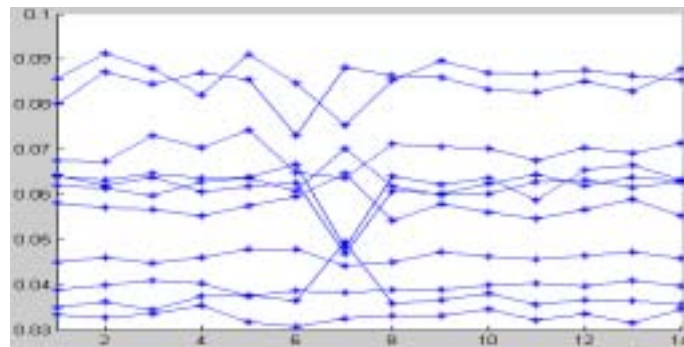


준에서의 큰 변동이 눈에 띈다. 실제로 각  $L$ 값에서의 정분류율은 (0.6867, 0.7090, 0.7200, 0.7323, 0.7305, 0.7382, 0.7397, 0.7502, 0.7602, 0.7668, 0.7818)이다. 이는  $L$ 이 클수록 정분류율이 커짐을 의미한다. 참고로 OPC와 EC로 실험을 한 경우에는 정분류율이 각각 0.4863, 0.7613으로 나타나는데  $L = 15$ 일 때 정분류율 0.7668과 비교하면  $L$ 의 증가가 단순히 정분류율의 증가를 의미하는 것은 아님을 알 수 있다.



<그림 5> Glass자료에서  $\sigma = 0.5$ 일 때 정분류율 평균

<그림 6> 에서 정분류율의 표준편차는 각각의  $L$ 에 대해서 (0.0862, 0.0843, 0.0694, 0.0574, 0.0624, 0.0614, 0.0627, 0.0462, 0.0395, 0.0373, 0.0331)인데  $L$ 이 증가할수록 감소하는 경향을 보임을 알 수 있다. OPC와 EC에는 코드행렬의 변동이 없기 때문에 표준편차가 0.0233, 0.0088로 작다.



<그림 6>  $\sigma = 0.5$ 일 때 정분류율 표준편차

<표 6>은 제안된 방법의 정분류율을 비교하기 위한 표이다. 모든 경우의 결과가 유사하기 때문에  $L = (6, 10, 15, 20)$ 의 경우를 고려하였고, 3, 7, 13번째의 기준 ( $A_{0.3}, A_1, L_{0.9}$ )의 결과를 나타내었다. <표 6>에서 알 수 있듯이, 제안된 방법의

정분류율은  $L$ 에 따라 조금씩 증가하고 표준편차는 감소하는 형태이고,  $\sigma=0.2$ 에서는 조금씩 증가하는 경우도 있다. 이는 분류기의 성능이 증가하면 본 논문에서 제안된 방법이 좀 더 효과적임을 증명해 주는 단적인 예이다. 괄호 안은 결합기에 분류기의 성능을 고려하지 않은 방법을 나타내는데, 제안된 방법보다 정분류율이 낮음을 알 수 있다. 그러나  $L$ 의 값이 크고  $\sigma=0.2$ 인 경우에는 정분류율이 유사함을 알 수 있다.

이상의 실험결과를 볼 때 Kuncheva(2005)에서 제안한  $L_{0.5}$ 기준 보다는  $A_{0.5}$ 기준이 표준편차면에서 조금 더 우수하다는 것을 알 수 있고,  $A_{0.5}$ 기준이  $L_{0.5}$ 기준보다 더 우수하다는 증거를 찾을 수 없었으며, 모든 기준의 정분류율에 유의한 차이를 발견할 수 없었다. 분류기의 개수가 증가함에 따라 정분류율도 증가하는 경향이 있지만 잘 설계된 코드행렬은 적은 개수의 분류기로도 높은 정분류율을 만들 수 있음을 알 수 있다. 각 분류기의 성능을 고려하는 결합기가 더 좋은 결과를 보인다는 것을 알 수 있다.

<표 6>  $L=(6, 10, 15, 20)$ 인 각 분류기의 정분류율

$L$	기 준	$\sigma=0.5$		$\sigma=0.2$	
		평 균	표준편차	평 균	표준편차
6	3	0.7057(0.6862)	0.0831(0.0881)	0.8251(0.7852)	0.1007(0.0865)
	7	0.7081(0.6967)	0.0855(0.0753)	0.8305(0.8176)	0.0920(0.0739)
	13	0.7050(0.6825)	0.0867(0.0864)	0.8251(0.7883)	0.1015(0.0875)
10	3	0.7530(0.7323)	0.0493(0.0598)	0.8833(0.8550)	0.0407(0.0684)
	7	0.7419(0.7243)	0.0607(0.0703)	0.8680(0.8579)	0.0556(0.0607)
	13	0.7532(0.7303)	0.0489(0.0616)	0.8833(0.8526)	0.0407(0.0669)
15	3	0.7794(0.7658)	0.0262(0.0344)	0.8973(0.7004)	0.0416(0.0317)
	7	0.7683(0.7589)	0.0468(0.0496)	0.8829(0.8849)	0.0618(0.0412)
	13	0.7794(0.7668)	0.0262(0.0364)	0.8973(0.8979)	0.0416(0.0358)
20	3	0.7802(0.7836)	0.0324(0.0335)	0.8988(0.8995)	0.0270(0.0402)
	7	0.7703(0.7758)	0.0341(0.0325)	0.8982(0.8737)	0.0291(0.0693)
	13	0.7802(0.7832)	0.0324(0.0316)	0.8988(0.8998)	0.0270(0.0338)

## 참고문헌

1. Aha, D. W. and Bankert, R. L. (1997). Cloud classification using error correcting output codes. *Artificial Intelligence Applications : Natural Resources, Agriculture and Environmental Science*, 11(1), 13-28.
2. Berger, A. (1999). Error-correcting output coding for text classification. *In Proceedings of Int. Joint Conf. Artificial Intelligence, IJCAI'99*, Stockholm, Sweden.
3. Breiman, L. (1997). Bagging predictors. *Machine Learning*, 24(2):123-140.
4. Dietterich, T. G. and Bakiri, G. (1991). Error-correcting output codes : A general method for improving multi-class inductive learning programs. *In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 572-577. AAAI Press.
5. Dietterich, T. G. and Bakiri, G. (1995). Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2, 263 -286.
6. Allwein, E. L., Schapire, R. E. and Singer, Y. (2000). Reducing multi-class to binary : A unifying approach for margin classifiers. *Machine learning research*, 1, 113-141.
7. Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and application to boosting. *Journal of computer and system science*, 55, 119 -139.
8. Kittler, J., Ghaderi, R., Windeatt, T. and Matas, G.(2001). *Face verification using error correcting output codes. In Computer Vision and Pattern Recognition CVPR01*, Hawaii, IEEE Press.
9. Kuncheva, L. I. and Whitaker, C. J.(2003). Measures of diversity in classifier ensembles. *Mach. Learn.* 51, 181-207.
10. Kuncheva, L. I. (2005) Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters* 26(1), 83 -90.
11. Schapire, R. E. (1997). Using output codes to boost multi-class learning problems. *In 14th International Conf. on Machine Learning*, pages 313-321. Morgan Kaufman.
12. Windeatt, T. and Ghaderi, R. (2003). Coding and decoding strategies for multi-class learning problems. *Information Fusion* 4(1), 11-21.

[ 2006년 4월 접수, 2006년 5월 채택 ]