The Korean Communications in Statistics **Vol. 14** No. 2, 2007, pp. 377–388

Comparison Study of Multi-class Classification Methods*

Whasoo Bae[1)] Gab Dong Jeon[2)] and Kyung Ha Seok[3)]

Abstract

As one of multi-class classification methods, ECOC (Error Correcting Output Coding) method is known to have low classification error rate. This paper aims at suggesting effective multi-class classification method (1) by comparing various encoding methods and decoding methods in ECOC method and (2) by comparing ECOC method and direct classification method. Both SVM (Support Vector Machine) and logistic regression model were used as binary classifiers in comparison.

Keywords: ECOC; encoding; decoding; classifier; multi-class classification.

1. 서론

분류는 훈련용 데이터 (training data) 에 대한 분석을 토대로 데이터의 클래스나 개념을 설명하고, 데이터의 클래스를 구별하는 모형들을 유도하여 새로운 데이터의 분류값을 예측하는 작업이다. 분류할 클래스가 2개인 이진분류인 경우에는 정확성과 효율성 등이 검증된 로지스틱 회귀모형, 의사결정나무, 신경망, SVM 등이 많이 사용되고 있다. 분류클래스가 3개 이상인 경우에는 이진분류 알고리즘을 다중분류의 경우로 확장하여 문제를 해결하는 방법과 하나의 분류기를 통하여 직접적으로 분류문제를 해결하는 방법이 있으나 문제해결에 있어 한계점을 가지고 있고, 이진분류기의 특정에 따라 다중분류기로 확장하는 것이 쉽지 않은 경우도 있다.

Dietterich와 Bakiri (1991, 1995) 는 직접적 분류방법의 한계를 개선하고 이진분류의 방법을 혼합시켜 다중분류문제에 잘 적용되도록 하는 ECOC (Error Correcting Output Code) 법을 제안하였다. ECOC법은 하나의 다중분류문제를 여러 개의 이진분류부문제 (binary classification sub-problem) 로 분할하여 해결한 후, 그 결과를 일정 기준에 근거하여 원래의 다중분류의 결과로 결합하는 방법이다. 이는 다중분류기에서 분류작업

* This work was supported by the 2006 Inje University Research Grant.

1) Associate Professor, Department of Data Science and Institute of Statistical Information, Inje University, Kimhae 621-749, Korea.
Correspondence : wbae@stat.inje.ac.kr
2) Researcher, Telluce Corporation, Seoul 135-951, Korea.
3) Professor, Department of Data Science and Institute of Statistical Information, Inje University, Kimhae 621-749, Korea.

을 한꺼번에 하는 것보다 상대적으로 작업이 용이한 이진분류기에서 작업을 시도한다는 점과 목표 값을 이진 값으로 바꾼 자료를 여러 번 반복훈련 함으로써 다중분류문제의 오류율을 낮추는 장점이 있다고 알려져 있다.

특히, Windeatt와 Ghaderi (2003) 는 효율적인 ECOC 설계방법과 다양한 디코딩방법에 대해 소개하였고, Kuncheva (2005) 는 코드행렬 설계에서 다양한 측도의 활용방안을 제시하고 제시된 방법에 대한 비교를 하였다. ECOC법은 패턴인식 분야에서 개발되어 최근에는 문자분류 등 많은 분야에서 사용되고 있다 (Aha과 Bankert, 1997; Berger, 1999; Kittler 등, 2001).

본 논문에서는 ECOC법을 적용한 다중분류에 사용되는 이진분류기의 종류와 코드행렬 설계에 관련되는 인코딩방법 및 디코딩방법을 비교하고 ECOC법과 직접적인 분류방법에 의한 다중분류결과를 비교하여 효율적인 방법에 대하여 제시하고자한다.

제 2장에서는 ECOC법의 코드행렬 설계방법인 인코딩방법과 이진분류부문제의 결과를 결합하는 방법인 디코딩방법에 대해서 소개하고, 제 3장에서는 실제 데이터에 적용하여 분류기로 로지스틱회귀모형과 SVM을 사용하여 인코딩 및 디코딩방법에 따른 분류정확도에 대해 비교하고, ECOC법과 직접적인 분류방법을 비교하였으며, 결론 및 향후과제에 대하여 제 4장에서 기술하였다.

2. ECOC법

ECOC법의 단계는 주어진 다중분류문제를 여러 개의 이진분류부문제로 변환하는 인코딩 (encoding) 단계와 이진분류 결과를 결합하여 원래의 클래스 값으로 환원하는 디코딩 (decoding) 단계로 나누어진다.

2.1. 인코딩

다중분류문제를 이진분류부문제로 변환하는 인코딩에서는 코드행렬의 설계가 중요한데, 대표적인 코드행렬 설계방법으로는 OPC (One Per Class) 방법, APC (All Pairwise Comparison) 방법 그리고 EC (Exhaustive coding) 방법을 들 수 있다.

OPC방법은 각 분류기가 하나의 클래스 값만 $\{1\}$로 하고 나머지는 $\{-1\}$로 하여 문제를 해결하는 방법으로써, 예들 들어 $c = 4$일 때 OPC방법의 코드행렬은 표 2.1과 같이 나타내게 된다.

코드행렬의 l번째 열을 분류기 D_l이라고 하고 코드행렬의 i번째 행을 코드워드 (code word) C_i라 하는데 표 2.1에서 원래 클래스 값이 $\{1\}$인 것은 각 분류기에서 나온 값이 $\{1, -1, -1, -1\}$로 변환됨을 의미한다.

APC방법은 가능한 모든 클래스의 쌍을 이용하는 방법으로써, 각 분류기에서 관심이 있는 두 개의 클래스의 값은 $\{1\}$또는 $\{-1\}$값을 가지고, 사용하지 않는 클래스는 $\{0\}$으로 표현하는 방법으로 $c = 4$일 때 APC방법의 코드행렬은 표 2.2와 같다.

EC방법은 c개의 클래스에 대하여 모든 가능한 $2^{(c-1)} - 1$경우의 문제로 나타내는 방법으로 $3 \le c \le 7$인 경우 EC방법의 코드행렬 설계방법은 다음과 같고 $c = 4$인 경우의

표 2.1: $c = 4$인 OPC 코드행렬

클래스 값	D_1	D_2	D_3	D_4
1	1	−1	−1	−1
2	−1	1	−1	−1
3	−1	−1	1	−1
4	−1	−1	−1	1

표 2.2: $c = 4$인 APC 코드행렬

클래스 값	D_1	D_2	D_3	D_4	D_5	D_6
1	1	1	1	0	0	0
2	−1	0	0	1	1	0
3	0	−1	0	−1	0	1
4	0	0	−1	0	−1	−1

EC방법의 코드행렬은 표 2.3과 같다.

1. 첫 번째 행은 모두 1이다.
2. 두 번째 행은 처음 $2^{(c-2)}$개는 −1, 나머지 $2^{(c-2)} - 1$개는 1의 순서로 구성된다.
3. 세 번째 행은 처음 $2^{(c-3)}$개는 −1, 다음 $2^{(c-3)}$개는 1, 그 다음 $2^{(c-3)}$개는 −1, 그리고 나머지 $2^{(c-3)} - 1$개는 1의 순서로 구성된다.
4. i번째 행은 $2^{(c-i)}$개의 −1과 1이 교대로 구성된다.
5. 마지막 행은 $-1, 1, -1, 1, -1, 1, \ldots, -1$순으로 나열된다.

표 2.3: $c = 4$인 EC 코드행렬

클래스 값	D_1	D_2	D_3	D_4	D_5	D_6	D_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

$8 \le c \le 11$의 경우, Dietterich와 Bakiri (1995) 는 최적화 방법을 통해 EC방법의 코드행렬로부터 열을 선택하는 방법을 권하고 있다. $c > 11$인 경우에 대해서는 랜덤코드생성이 효과적이라고 나타나 있으며, 코드워드의 랜덤결정법의 효과에 대해서는 많은 연구결과들이 보여주고 있다 (Dietterich와 Bakiri, 1995; Schapire, 1997; Windeatt과 Ghaderi, 2003; Kuncheva, 2005).

코드행렬을 설계함에 있어 코드워드간의 거리인 행간 거리와 분류기간의 거리인

열간 거리를 고려해야 하는데, 가장 보편적으로 사용되는 것이 해밍거리 (Hamming Distance) 이다. 코드워드 C_i와 C_j의 해밍거리 (HC_{ij})는 $C(i,k)$를 C_i의 k번째 값이라 할 때

$$HC_{ij} = \sum_{k=1}^{L} \frac{|C(i,k) - C(j,k)|}{2}, \quad i,j = 1,2,\ldots,c \tag{2.1}$$

로 나타내는데 HC_{ij}값이 커지도록 코드행렬을 설계함으로써 이진분류기에서 발생하는 오류를 극복할 수 있다.

코드행렬의 설계는 코드워드간의 거리뿐만 아니라 각각의 분류기들의 독립성을 의미하도록 분류기간의 거리도 크게 해야 하는데 분류기 D_i와 D_j의 해밍거리 (HD_{ij})는 다음과 같이 나타낸다.

$$HD_{ij} = \min_{i,j,i\neq j} \left\{ \sum_{k=1}^{c} \frac{|C(k,i) - C(k,j)|}{2}, \ \sum_{k=1}^{c} \frac{|2 - C(k,i) - C(k,j)|}{2} \right\}, \quad i,j = 1,\ldots,L \tag{2.2}$$

코드행렬을 설계할 때 코드워드간의 거리와 분류기간의 거리는 최대로 하는 것이 목표가 되는데, 코드행렬의 불일치를 측정하는 기준으로 코드워드간 거리와 분류기간 거리에 대하여 각각의 평균거리를 이용한 A기준과 이들에 대한 각각의 최소거리를 이용한 L기준을 일반적으로 이용한다.

HC_{ij}와 HD_{ij}의 평균을 각각 AHC와 AHD는 다음과 같이 나타내고

$$AHC = \frac{2}{c(c-1)} \sum_{i=1}^{c} \sum_{i<j}^{c} HC_{ij}/L, \tag{2.3}$$

$$AHD = \frac{2}{L(L-1)} \sum_{i=1}^{L} \sum_{i<j}^{L} HD_{ij}/c, \tag{2.4}$$

HC_{ij}와 HD_{ij}의 최소값을 각각 LHC와 LHD라 하여 아래와 같이 정의한다.

$$LHC = \min(HC_{ij}), \ i,j = 1,\ldots,c, \tag{2.5}$$

$$LHD = \min(HD_{ij}), \ i,j = 1,\ldots,L. \tag{2.6}$$

AHC와 AHD의 가중평균인 A_α와 LHC와 LHD의 가중평균을 나타내는 L_α를

$$A_\alpha = \alpha AHC + (1-\alpha)AHD, \quad 0 \le \alpha \le 1, \tag{2.7}$$

$$L_\alpha = \alpha LHC + (1-\alpha)LHD, \quad 0 \le \alpha \le 1 \tag{2.8}$$

로 각각 나타내고 코드워드간의 거리와 분류기간의 거리를 고려함에 있어 가중치 α값의 변화에 따른 행간거리와 열간거리에 대한 성능을 비교할 수 있다. 3장에서는 A_α와 L_α기준을 적용하여 설계한 코드행렬과 잘 알려진 코드행렬과의 비교를 통해 어떤 기준이 더 우수한지 실험을 통해 살펴본다.

2.2. 디코딩

검증용 데이터 $\mathbf{x}$가 입력이 되면, L개의 이진분류기에서 출력값, $\hat{\mathbf{y}}=(\hat{\mathbf{y}}_1,\hat{\mathbf{y}}_2,\ldots,\hat{\mathbf{y}}_L)$이 계산되고 출력값 $\hat{\mathbf{y}}$과 각 코드워드간의 거리를 계산하여, 가장 작은 값에 해당하는 코드워드의 클래스에 $\mathbf{x}$를 분류하게 되는데 이 단계를 디코딩이라고 한다. 실수 또는 이진형 값을 가지지는 출력값 $\hat{\mathbf{y}}$을 원래의 클래스 값으로 환원시키는 결합기 (ensemble) 에서는 $\hat{\mathbf{y}}$과 각 코드워드간의 거리를 어떻게 정의하느냐가 매우 중요하며, 이에 따라 여러 가지 방법이 고려될 수 있다.

해밍 디코딩 (Hamming Decoding) 방법에서는 이진형 출력값 $\hat{\mathbf{y}}$의 라벨 $\mathbf{R}$은

$$R = \underset{i}{\operatorname{argmin}} \left\{ \sum_{k=1}^{L} |\operatorname{sgn}(y_k) - C(i,k)| \right\}, \ i = 1, \ldots, c \tag{2.9}$$

로 나타내진다. 단, 해밍거리가 동일한 값을 가지는 경우에는 해당 클래스 값 중에서 랜덤하게 분류를 하게 되는데 이는 나머지 디코딩방법에서도 동일하게 적용된다.

선형 디코딩 (linear decoding) 방법에서의 출력값 $\hat{\mathbf{y}}$의 라벨 R은 다음과 같다.

$$R = \underset{i}{\operatorname{argmin}} \left\{ \sum_{k=1}^{L} |y_k - C(i,k)| \right\}, \ i = 1, \ldots, c. \tag{2.10}$$

해밍 디코딩이 단지 부호가 가장 많이 일치하는 코드워드의 클래스로 분류하는 반면, 선형 디코딩은 산술적으로 가까운 코드워드의 클래스로 분류를 하게 된다.

지수 디코딩 (exponential decoding) 방법에서는 출력값 $\hat{\mathbf{y}}$의 라벨 R이

$$R = \underset{i}{\operatorname{argmin}} \left\{ \sum_{k=1}^{L} \exp(-y_k C(i,k)) \right\}, \ i = 1, \ldots, c \tag{2.11}$$

로 나타낼 수 있다. 선형 디코딩과 달리 지수함수를 사용하면 거리가 가까울수록 0에 가까운 값을 가지고 거리가 멀수록 차이가 더욱 커짐으로써 보다 정확한 분류를 할 수 있다.

양선형 디코딩 (positive linear decoding) 방법은 출력값 $\hat{\mathbf{y}}$의 라벨 R은

$$\begin{aligned} R = \underset{i}{\operatorname{argmin}} \sum_{k=1}^{L} (1 - y_k C(i,k)) \times \mathrm{I}((1 - y_k C(i,k)) > 0), \\ k = 1, \ldots, L, \ i = 1, \ldots, c \end{aligned} \tag{2.12}$$

이며, 여기서 $\mathrm{I}(\cdot)$는 $(\cdot)$이 참이면 1, 그렇지 않을 경우 0의 값을 가지는 함수를 나타낸다.

3. 실험

이 장에서는 자료를 이용하여 ECOC법에서의 다양한 인코딩방법과 디코딩방법을 사용하여 ECOC법과 직접적 분류방법에서의 오분류율을 비교하였으며 사용된 자료는

표 3.1: 데이터 요약((http://www.ics.uci.edu/ mlearn/databases)

Data Set	Number of Examples		Number of Attributes	Number of Classes
	Train	Test		
glass	214	·	9	7
ecoli	336	·	8	8
dermatology	366	·	34	6
vowel	528	462	10	11

표 3.1에 요약 되어있다. 표 3.1의 자료중, dermatology, ecoli, glass데이터는 검증용 자료가 없는 훈련용 자료만이 주어졌기 때문에 훈련용 자료와 검증용 자료의 비율을 7:3으로 한 교차확인을 50회 실시하여 실험을 수행하고 얻어진 오분류율의 평균을 구하여 비교하였다. 검증용자료가 주어진 vowel 자료는 클래스의 수가 11개로 분류가 힘든 자료로 자주 이용되는 자료다.

3.1. ECOC법의 분류정확도에 대한 비교

ECOC법의 인코딩 단계의 코드행렬 설계방법과 디코딩 단계의 결합기 설계방법에 따른 분류정확도의 비교를 위해 코드행렬은 OPC와 APC 그리고 식 (2.7) 과 식 (2.8) 의 기준을 적용한 $L_1, L_{0.5}, A_{0.5}$를 사용하였고, 결합기는 해밍 디코딩, 선형 디코딩, 지수 디코딩, 양선형 디코딩 방법을 사용하였다. 이진분류기로는 SVM과 로지스틱 회귀모형을 사용하여 분류기 종류에 따른 분류정확도의 차이를 알아보았으며 실험 결과는 표 3.2에 나타나 있다. 그림 3.1과 그림 3.2는 표 3.2의 결과를 그림으로 나타내고 있다.

표 3.2과 그림으로부터 SVM을 분류기로 사용한 ECOC (이하 ECOC-SVM으로 표현함) 의 경우가 로지스틱 회귀모형을 분류기로 사용한 ECOC (이하 ECOC-Logistic으로 표현함) 의 경우보다는 오분류율이 일반적으로 낮음을 알 수 있다.

인코딩단계에서의 방법을 비교해 보면 ECOC-SVM에서는 glass 데이터와 ecoli 데이터에서 $A_{0.5}$기준의 코드행렬의 결과가 가장 좋고 APC 코드행렬도 그에 준하는 좋은 결과를 보였다. dermatology 데이터와 vowel 데이터에서는 APC 코드행렬의 결과가 가장 좋다는 것을 알 수 있다. ECOC-Logistic의 경우에는 모든 데이터에서 APC 코드행렬을 적용하였을 때 가장 낮은 오분류율을 보임을 알 수 있다. 반면, 두 경우 다 OPC를 코드행렬로 사용하였을 때 오분류율이 가장 높게 나왔다 APC 코드행렬은 설계에서부터 이진분류부문제로 변환함에 있어 관심 있는 클래스만을 사용하도록 설계되었고, $A_{0.5}$기준의 코드행렬은 행간 거리와 열간 거리의 최대값에 대한 평균을 고려하여 설계된 것이므로, 잘 설계된 만큼 다중분류문제에 있어 높은 분류정확도를 얻음을 보여준다.

디코딩 방법에 따른 결과를 보면, 거리의 측도로 부호의 일치여부만을 고려한 해밍 디코딩 방법이 가장 결과가 좋지 않았으며, 나머지 방법에 대해서는 약간의 차이는 있지만 거의 유사한 결과를 볼 수 있다. 이는 거리의 측도로 부호뿐만 아니라 근접정도를 고려한 디코딩 방법인 선형, 지수, 양선형 디코딩 방법이 보다 좋은 결과를 나타냄을 알

수 있다.

ECOC법을 적용하여 다중분류문제를 해결함에 있어 일반적으로 분류기로 SVM을 사용하고 APC 코드행렬과 근접정도를 고려한 디코딩 방법을 쓰는 것이 일반적으로 보다 좋은 결과를 얻는다고 볼 수 있다.

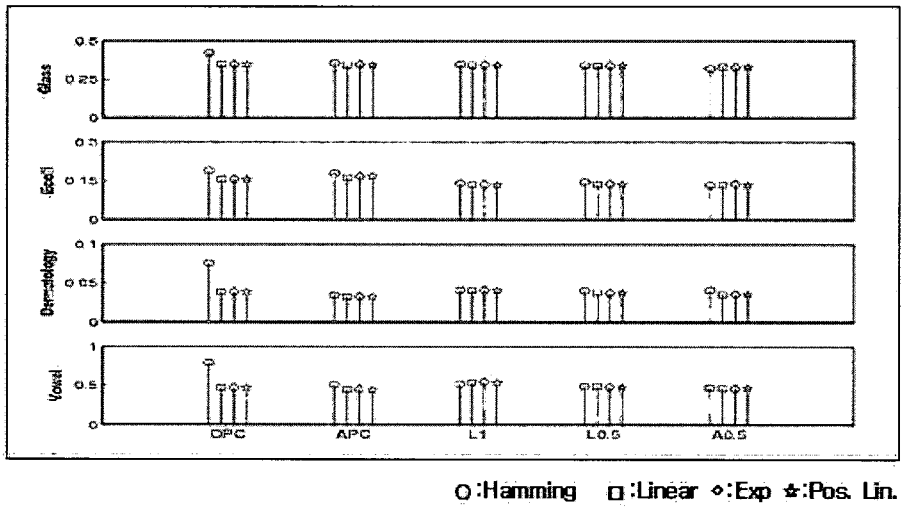

그림 3.1: 인코딩방법 및 디코딩방법에 따른 오분류율 (ECOC-SVM)

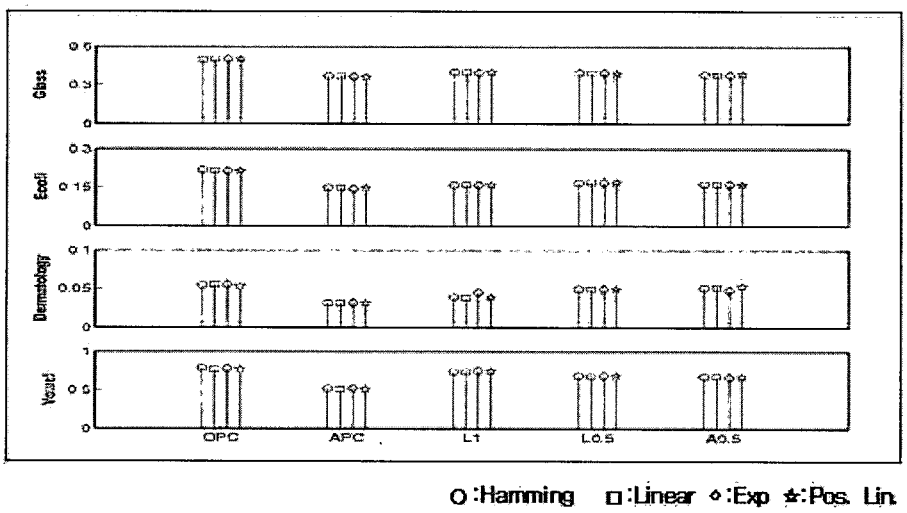

그림 3.2: 인코딩방법 및 디코딩방법에 따른 오분류율 (ECOC-Logistic)

표 3.2: 인코딩 방법 및 디코딩 방법에 따른 오분류율

		오분류율 (위: ECOC-SVM, 아래:ECOC-Logistic)			
사용된 자료	코드 행렬	디코딩 단계의 결합기 설계방법			
		Hamming	Linear	Exp.	Pos. Lin.
Glass	OPC	0.4213	0.3463	0.3466	0.3466
		0.4991	0.5031	0.5003	0.5022
	APC	0.3550	0.3417	0.3459	0.3414
		0.3728	0.3719	0.3669	0.3644
	L_1	0.3469	0.3416	0.3422	0.3400
		0.4019	0.4003	0.3978	0.4009
	$L_{0.5}$	0.3409	0.3369	0.3378	0.3378
		0.3978	0.3963	0.3969	0.3941
	$A_{0.5}$	0.3200	0.3306	0.3281	0.3300
		0.3828	0.3803	0.3781	0.3853
Ecoli	OPC	0.1912	0.1576	0.1574	0.1573
		0.2186	0.2172	0.2136	0.2158
	APC	0.1810	0.1646	0.1686	0.1693
		0.1500	0.1494	0.1444	0.1492
	L_1	0.1410	0.1364	0.1376	0.1362
		0.1588	0.1600	0.1594	0.1592
	$L_{0.5}$	0.1470	0.1390	0.1390	0.1380
		0.1686	0.1706	0.1680	0.1698
	$A_{0.5}$	0.1342	0.1358	0.1382	0.1354
		0.1628	0.1632	0.1618	0.1620
Dermatology	OPC	0.0761	0.0391	0.0391	0.0391
		0.0546	0.0557	0.0553	0.0533
	APC	0.0350	0.0327	0.0327	0.0327
		0.0314	0.0314	0.0314	0.0314
	L_1	0.0411	0.0407	0.0411	0.0407
		0.0394	0.0381	0.0452	0.0387
	$L_{0.5}$	0.0404	0.0378	0.0374	0.0376
		0.0499	0.0499	0.0495	0.0497
	$A_{0.5}$	0.0411	0.0357	0.0357	0.0359
		0.0510	0.0520	0.0482	0.0531
Vowel	OPC	0.7987	0.4784	0.4740	0.4740
		0.7900	0.7771	0.7835	0.7792
	APC	0.5141	0.4535	0.4562	0.4459
		0.5216	0.5108	0.5238	0.5130
	L_1	0.5238	0.5390	0.5476	0.5390
		0.7424	0.7446	0.7554	0.7511
	$L_{0.5}$	0.4913	0.4913	0.4848	0.4827
		0.6926	0.6775	0.6905	0.6905
	$A_{0.5}$	0.4762	0.4654	0.4654	0.4719
		0.6775	0.6797	0.6688	0.6753

3.2. ECOC법과 직접적 분류방법간의 분류정확도

Lee 등 (2004) 는 이진클래스의 데이터만을 분류할 수 있도록 설계된 알고리즘인 SVM을 보완하여 다중분류문제를 직접 분류하는 MSVM (Multicategory Support Vector Machine) 을 제안했다.

이 장에서는 다중분류문제 해결에 있어서 ECOC-SVM과 직접적으로 분류하는 MSVM에 대한 분류정확도를 비교하고 또 ECOC-Logistic과 로지스틱 회귀모형에 의한 직접적 분류에서의 분류정확도를 비교하기로 한다. ECOC-SVM과 ECOC-Logistic의 결과는 OPC를 제외한 나머지 코드행렬과 해밍 디코딩방법외의 디코딩 방법의 결과 중에서 분류정확도가 가장 좋은 값을 선정하여 비교하였다. 표 3.3과 표 3.4에 ECOC-SVM과 MSVM, ECOC-Logistic과 로지스틱 회귀모형에 의한 오분류율을 각각 나타내고 그림 3.3은 두 결과를 종합하여 나타내었다.

표 3.3으로부터 ECOC-SVM과 MSVM을 적용한 경우에는 glass 데이터와 ecoli 데이터에 있어서 ECOC-SVM의 오분류율이 낮으며, dermatology 데이터는 MSVM의 결과가 좋게 나타남을 알 수 있다. vowel 데이터의 경우는 많은 클래스의 수로 인해 MSVM 수행과정에서 메모리가 부족하여 실험에 대한 결과를 도출하지 못하였다. dermatology 데이터를 제외한 glass 데이터와 ecoli 데이터에서 ECOC-SVM을 이용한 방법이 더 나은 결과를 보였고, MSVM은 데이터의 클래스수가 많을 경우 계산이 불가능하다는 단점이 있으므로 다중분류문제를 해결하기 위하여 분류기로 SVM을 이용할 경우에는 ECOC법을 이용하는 것이 더 효율적인 방법이라 할 수 있다.

표 3.4로부터 ECOC-Logistic과 로지스틱 회귀모형의 결과를 보면 모든 데이터에 대하여 ECOC-Logistic이 분류정확도가 높게 나타났다. 로지스틱 회귀모형을 이용하여 직접적으로 분류하는 경우가 계산량과 계산시간이 적으나 분류정확도만을 고려한 본 실

표 3.3: ECOC-SVM과 MSVM 오분류율

Data	ECOC-SVM	MsVM
glass	0.3281	0.3458
ecoli	0.1354	0.1548
dermatology	0.0327	0.0279
vowel	0.4535	·

표 3.4: ECOC-Logistic과 Logistic 오분류율

Data	ECOC-Logistic	Logistic
glass	0.3644	0.3852
ecoli	0.1444	0.1550
dermatology	0.0314	0.0682
vowel	0.5108	0.5130

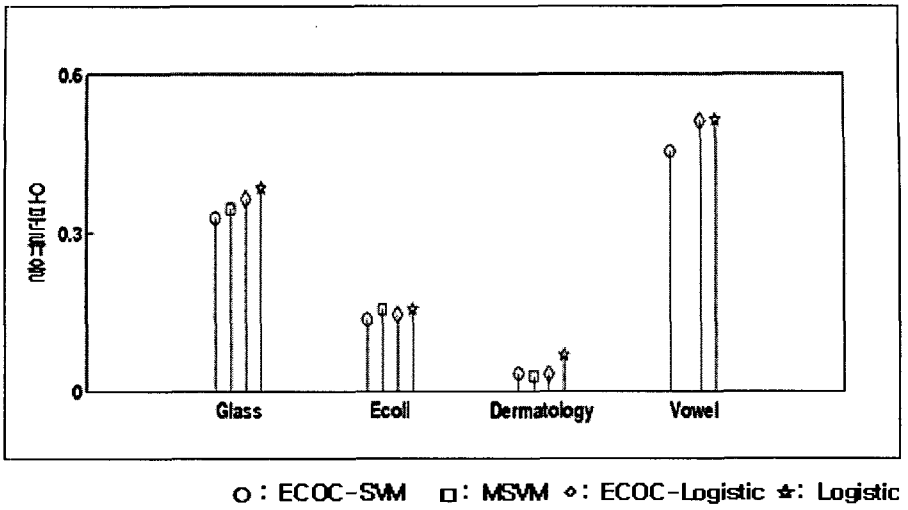

그림 3.3: ECOC와 직접적 분류방법에 따른 오분류율

험에서는 로지스틱 회귀모형을 이용한 직접적인 분류방법보다 ECOC법을 적용한 방법이 결과가 더 좋음을 알 수 있다.

그림 3.3에 나타난 종합적인 결과로부터 ECOC-SVM을 적용한 경우의 오분류율이 가장 낮음을 알 수 있다. 따라서 잘 설계된 코드행렬로 APC를 사용하고 근접정도까지 고안된 적절한 디코딩방법을 선택하여 적용시킨 ECOC-SVM이 다중분류문제를 해결함에 있어서는 효율적이라고 할 수 있다.

4. 결론

다중분류문제를 해결하기 위하여 제안된 ECOC법의 인코딩 단계에서는 코드행렬의 설계가 가장 중요하므로, 코드행렬을 설계하는 데 있어서 코드워드간의 최소거리, 분류기간 거리의 변동성, 분류기의 수 그리고 분류기의 독립성 등은 꼭 고려되어야 할 사항이다. 또한, 디코딩 단계에서는 인코딩 단계를 거친 이진 출력값을 최소거리에 해당하는 클래스로의 분류를 위한 거리에 대한 정의가 매우 중요하다.

ECOC법을 적용한 실험 결과로부터 인코딩 단계에서의 코드행렬에 따른 분류정확도를 살펴보면, ECOC-SVM과 ECOC-Logistic 결과 모두 코드행렬로 OPC를 사용하였을 때의 분류정확도가 가장 낮았으며, 자료와 분류기에 따라 다소의 차이는 있지만 일반적으로 APC 코드행렬과 $A_{0.5}$기준의 코드행렬이 전반적으로 좋은 분류정확도를 나타내었다. 특히 APC 코드행렬은 설계가 특별히 필요없고 계산시간이 적게 들면서 분류결과가 좋게 나타났다. 디코딩 단계에서는 부호만 고려한 해밍 디코딩 방법이 분류정확도가 가장 낮았고 나머지 디코딩 방법들은 거리의 측도로 부호뿐만 아니라 근접정도를 고려한 디코딩 방법으로 보다 좋은 결과를 나타냄을 알 수 있었다.

동일한 코드행렬과 디코딩방법을 이용하여 ECOC법을 적용한 분류기의 비교에서는 SVM을 분류기로 사용하는 경우가 일반적으로 나은 분류결과를 보였다.

ECOC법과 직접적 분류방법간의 분류정확도를 비교한 결과, SVM을 분류기로 사용하는 경우가 로지스틱모형을 사용한 경우보다 나은 결과를 보였고 특히 MSVM보다 ECOC_SVM이 그리고 ECOC-Logistic이 로지스틱 직접분류보다 나은 결과를 보였다. 특히 MSVM의 경우엔 한꺼번에 직접적으로 분류를 해야 하므로 계산해야 하는 양이 많아 시간이 오래 걸리는 반면, ECOC_SVM은 MSVM보다 상대적으로 적은 계산시간으로 만족할 만한 높은 분류정확도를 보였다. 또한, MSVM은 데이터의 클래스수가 많을 경우 계산이 불가능하다는 단점이 있으므로 다중분류문제를 해결하기 위하여 분류기로 SVM을 이용할 경우에는 ECOC법을 이용하는 것이 더 효율적인 방법이라 할 수 있다. 잘 설계된 코드행렬과 명확한 거리 측도에 의한 디코딩방법을 적용한 ECOC법은 다중분류문제를 해결함에 있어 사용하기에 좋은 성질을 가지고 있음을 알 수 있었다.

본 논문에서는 프로그램 수행상의 한계로 SVM과 로지스틱모형만을 분류기로 비교하였으나 보다 많은 분류기법과 비교하는 작업으로 ECOC법에 대한 보다 보편적인 결과를 얻어낼 필요가 있다고 생각된다. 또한, 이진분류기를 통한 결과를 결합하여 다중분류로 환원하는 결합기를 설계함에 있어서 근접정도를 나타내는 측도를 계산하기위한 효율적인 방법에 대한 지속적인 연구가 요구된다.

참고문헌

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**, 13–28.

Berger, A (1999). Error-correcting output coding for text classification. *In Proceedings of International Joint Conference Artificial Intelligence, IJCAI'99,* Stockholm, Sweden.

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),* 572–577, AAAI Press.

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.

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.

Kuncheva, L. I. (2005). Using diversity measures for generating error -correcting output codes in classifier ensembles. *Pattern Recognition Letters,* **26**, 83–90.

Lee, Y., Lin Y. and Wahba, G. (2004). Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association,* **99,** 67–81.

Schapire, R. E. (1997). Using output codes to boost multi-class learning problems. *In 14th International Conference on Machine Learning,* 313–321, Morgan Kaufman.

Windeatt, T. and Ghaderi, R. (2003). Coding and decoding strategies for multi-class learning problems. *Information Fusion,* **4**, 11–21.

[Received April 2007, Accepted June 2007]