

지지벡터기계를 이용한 다중 분류 문제의 학습과 성능 비교

황 두 성[†]

요 약

이진 분류기로서 지지벡터기계는 다양한 응용을 통해 이진 분류 문제에서 기존의 패턴 분류기들보다 우수한 성능을 보였다. 지지벡터기계의 바탕이 되는 최대 마진 분류 이론을 다중 분류 문제에 확장은 어려움이 있다. 이 논문에서는 다중 분류 문제를 위한 지지벡터기계의 학습 전략을 논의하였으며 성능 비교를 수행하였다. 학습 데이터의 분배 전략에 따라 지지벡터기계는 고유의 이진 분류 특징을 수정하지 않고 다중 분류 문제에 쉽게 적용될 수 있다. 다양한 벤치마킹 데이터에 대해 선택된 학습 전략, 커널함수, 학습 소요 시간 등에 따라 성능비교가 수행되었고 오류역전파 학습의 신경망의 테스트 결과와 비교되었다. 신경망 모델과 비교 실험에서 지지벡터기계는 일반적인 다중 분류 문제에 응용성과 효과가 있음을 보였다.

Learning and Performance Comparison of Multi-class Classification Problems based on Support Vector Machine

Doosung Hwang[†]

ABSTRACT

The support vector machine, as a binary classifier, is known to surpass the other classifiers only in binary classification problems through the various experiments. Even though its theory is based on the maximal margin classifier, the support vector machine approach cannot be easily extended to the multi-classification problems. In this paper, we review the extension techniques of the support vector machine toward the multi-classification and do the performance comparison. Depending on the data decomposition of the training data, the support vector machine is easily adapted for a multi-classification problem without modifying the intrinsic characteristics of the binary classifier. The performance is evaluated on a collection of the benchmark data sets and compared according to the selected learning strategies, the training time, and the results of the neural network with the backpropagation learning. The experiments suggest that the support vector machine is applicable and effective in the general multi-class classification problems when compared to the results of the neural network.

Key words: multi-class classification(다중 분류), support vector machine(지지벡터기계), class decomposition(클래스 분배), discrimination function(분리 함수)

21. 서 론

기계 학습 알고리즘을 현실적 분류 문제에서 응용

시 측정 가능한 데이터의 양은 제한되며 모 집단의 사전 정보가 부족하다. 선택된 학습 모델의 파라미터들을 학습하는데 준비된 데이터를 가지고 추정하는

※ 교신저자(Corresponding Author): 황두성, 주소: 천안시 안서동 산 29번지(330-714), 전화: 041)550-3485, FAX: 041)550-3480, E-mail: dshwang@dankook.ac.kr
접수일: 2007년 10월 25일, 완료일: 2008년 4월 28일

[†] 단국대학교 컴퓨터학과

※ 본 연구는 2006년 단국대학교 대학연구비의 지원으로 연구되었습니다.

경험적 학습 모델이 높은 일반화 성능을 제시하기 위해서는 데이터 전처리, 학습, 테스트와 평가의 일련의 과정을 반복하게 된다. 학습 파라미터들의 설정이 까다로운 신경망, 통계 모델 등에 비해 지지벡터 기계(Support Vector Machine, SVM)는 모델 파라미터의 설정이 용이하나 이진 분류기의 특성을 가지고 있어 다중 분류 문제의 적용에 단점이 있다[1,2].

다중 분류(multi-classification) 문제의 해결에 오류 역전파 학습 알고리즘(backpropagation learning algorithm,[3]) 또는 RBF 학습 알고리즘[1]의 적용 시 입력층의 뉴런의 수는 학습 데이터의 차원과 같으며, 출력층 뉴런의 수는 목적 클래스의 수와 같게 구성된다. 중간층의 수와 각 뉴런의 수는 주어진 분류 문제의 복잡도에 따라 학습 실험을 통한 시도로 결정하게 된다. 지금까지 연구된 결과로부터 학습 복잡도는 준비된 학습 데이터의 수, 목적 클래스의 수, 학습 데이터의 숨겨진 구조에 의존하기 때문에 다중 신경망의 토폴로지(topology)의 선택이 어렵다[3,4]. 그러나 이진 분류 문제에 다중 신경망의 적용은 다중 분류 문제보다는 복잡도와 토폴로지의 선택이 단순하다.

이진 분류기의 다중 분류 문제에 적용은 각 목적 클래스에 속한 데이터를 재구성하여 여러 개의 이진 분류기를 학습시킨다[5-7]. 각 이진 분류기의 성능은 학습에 사용된 데이터를 가지고 측정되며 테스트 데이터는 모든 학습된 이진 분류기의 입력으로 주어져 각각 예측된 결과를 조합하여 성능을 평가한다. 그러나 UCI 기계 학습 데이터의 Yeast와 같이 학습 데이터가 여러 개의 목적 클래스에 중복하여 나타나는 다중 분류 문제에서는 각 이진 분류기의 학습 데이터의 구성은 단지 하나의 클래스에만 속하게 재구성하여야 한다[4]. 그렇지 않으면 이러한 문제에 이진 분류기를 사용할 경우 여러 목적클래스에 속하는 학습 데이터의 학습 방해로 인해 학습이 어려워 높은 일반화 성능을 기대하기 어렵다. 주어진 학습 데이터의 분석에서 이러한 데이터 중복 문제를 파악하고 목적 클래스에 따라 학습 데이터를 재구성하여 다수의 이진 분류기를 학습하는 전략이 효과적이다[8-10].

본 논문에서는 이진 분류 문제에서 우수한 성능을 보이는 지지벡터기계를 다중 분류기로 확장하는 학습 전략을 살펴보고 JNI(Java Native Interface)를 이용한 지지벡터기계 기반 다중 분류 학습기를 구현하였다. 그리고 공개된 분류 문제를 가지고 다중 학습

기법에 따라 학습과 테스트 성능을 평가하였다. 2절에서 다중 분류 문제의 이진 분류 문제로 변형하는 학습 전략과 클래스 예측 방법을 논의하며 3절에서는 지지벡터기계의 학습에 대해 살펴보고 SVM^{light}을 이용한 다중 분류를 위한 지지벡터기계의 구현을 소개한다. 4절에서는 공개된 다중 분류 문제에서 구현된 지지벡터기계의 학습 전략 별 테스트 결과를 제시하고 다중 분류 신경망의 결과와 비교한다. 마지막으로 5절에서 데이터의 재구성에 기반을 둔 지지벡터기계 알고리즘을 이용한 다중 분류기의 응용에서 나타나는 문제를 지적하고 결론을 맺는다.

2. 관련연구

이진 분류기를 이용한 K -클래스 분류 문제(K -class classification problem, $K > 2$)의 학습은 주어진 분류 문제를 K 개 이상의 단순 이진 분류 문제로 쪼개어 학습시킨 후 조합하여 본래 문제를 해결한다. 각 이진 분류기는 주어진 입력벡터가 클래스의 멤버인지 아닌지를 변별하도록 학습된다. 테스트의 예측 클래스는 이진 분류기의 구성에 따른 다양한 방법으로 각 학습기의 출력을 조합하여 예측된다. 주어진 K -클래스 분류 문제 S 를 다중 분류 알고리즘 M 이 학습하면 테스트 벡터 x 의 분류예측은 식 (1)과 같다. L 은 학습벡터의 인덱스, x_i 는 학습벡터이고 y_i 는 x_i 의 클래스이다.

$$S = \{ \langle x_i, y_i \rangle \mid x_i \in R^n, y_i = k, i = 1, \dots, L, k = 1, \dots, K \} \quad (1)$$

$$M(x) = c, c = 1, 2, \dots, K$$

K -클래스 분류 문제에 대한 단일 학습(one-against-all learning, OVA) 기법은 입력 벡터가 클래스의 멤버인지를 예측하는 K 개의 이진 분류기 $M_k, k=1, 2, \dots, K$ 를 구성시킨다[6,7,9]. M_k 의 학습 데이터 S_k 는 클래스 k 에 속하는 학습 벡터가 긍정 클래스(positive class, $y=+1$), 그렇지 않으면 부정 클래스(negative class, $y=-1$)로 하는 식 (2)와 같이 구성시킨다.

$$S_k = \{ \langle x_i, +1 \rangle \mid \langle x_i, y_i \rangle \in S, y_i = k \} \cup \{ \langle x_i, -1 \rangle \mid \langle x_i, y_i \rangle \in S, y_i \neq k \}$$

$$M(x) = \arg \max_k M_k(x), k = 1, 2, \dots, K \quad (2)$$

테스트 벡터 x 는 모든 학습된 이진 분류기에 입력되어 가장 큰 출력값을 낸 분류기 M_k 의 인덱스 k 를 예측

클래스가 되는 승자독식(winer-take-all)을 사용한다. 이러한 다중 분류 학습 전략이 갖는 단점은 큰 K 값에 대해 학습 데이터의 분배로 인한 각 학습기는 부정 클래스에 속하는 벡터가 많아지는 데이터 불균형 문제 (data imbalanced problem,[11])가 발생할 수 있다.

쌍별 학습(all-pairs learning, AP)은 K-클래스 분류 문제에 대해 가능한 쌍별 $K(K-1)/2$ 이진 분류기들을 학습시킨다[5,8,10]. 클래스 i 와 j 를 분류하는 이진 분류기 $M_{ij}(i,j=1,2,\dots,K, i < j)$ 는 클래스 i 의 학습 벡터를 긍정 클래스로 클래스 j 에 속한 학습 벡터를 부정 클래스로 하는 학습 데이터 S_{ij} 를 학습하며 식 (3)과 같다.

$$S_{ij} = \{ \langle x_i, y_i \rangle \mid \langle x_i, y_i \rangle \in S_{y_i = i} \} \cup \{ \langle x_j, y_j \rangle \mid \langle x_j, y_j \rangle \in S_{y_j = j} \}$$

$$M(x) = \begin{cases} k & \text{if } h_{kc}(x) = k-1, h_{kc}(x) = \sum_{c=k+1}^K M_{kc}(x) \\ K & \text{otherwise} \end{cases} \quad (3)$$

테스트 벡터 x 에 대해 학습기의 결과로부터 $h_{kc}(x)=k-1$ 인 k 를 예측 클래스가 되는 다수결 투표 (majority vote)에 따른다. 쌍별 학습 전략을 이용한 분류기는 단지 두 클래스에 속한 벡터들에 대해 학습이 되어 단일 학습 전략에 비해 빠른 학습이 진행되나, 큰 K 값에 대해 학습기의 수가 $O(K^2)$ 로 증가하게 된다. 단일 학습 학습전략에서 학습데이터의 재구성 시 나타나는 부정클래스에 속하는 벡터가 많아지는 데이터 불균형문제는 고려하지 않아도 된다.

일반적으로 적용될 수 있는 다중 분류 학습 방법에는 코드 행렬(coding matrix)을 사용한다[7]. 선정된 코드 행렬 $D \in \{-1, 0, +1\}^{K \times L}$ 의 열에 제시된 분류 정보에 따라 L개의 이진분류기 $M_l, l=1, 2, \dots, L$ 를 학습한다. l-열은 분류기 M_l 을 학습 데이터의 구성에 필요한 긍정과 부정 클래스의 정보를 가지고 있다. 단일 학습의 경우 코드 행렬의 구성은 $L=K$ 이며 l-열은 M_l 을 학습하는데 필요한 코드 정보 $D_{ji}=+1, D_{ji}=-1(i=1, 2, \dots, j-1, j+1, \dots, K)$ 가 된다. 코드 정보에 따라 클래스 j 에 속하는 벡터는 긍정 클래스가 되며 그 외는 부정클래스로 데이터를 재구성하여 M_l 를 학습한다. 쌍별 학습에서 $L=K(K-1)/2$ 이며 M_l 을 학습하는데 필요한 코드정보는 $D_{ii}=+1, D_{ji}=-1, D_{ji}=0, k=1, 2, \dots, K(k \neq i \neq j)$ 가 된다. 코드 행렬을 이용한 이진분류기의 확장은 클래스 사이에 내재된 경험적 분류 지식이 제공되면 계층적 다중 분류 학습을 구성시킬 수 있는 유연성을 제공할 수 있다. 실험에 따르면

코드 행렬과 예측된 결과의 손실 함수(loss function) 관계는 최종 예측 결과를 보정하여 높은 분류율을 제공하였다. 이러한 보정기법에는 평균이진손실 (average binary loss), Hamming decoding, loss-based decoding 등이 알려져 있다[7,8].

3. 지지벡터기계 기반 다중 분류 학습

3.1 지지벡터기계 알고리즘

두 클래스로 구성된 선형 분리가 가능한 문제 $S = \{ \langle x_i, y_i \rangle \mid x_i \in R^n \text{ and } y_i = \pm 1, i=1, 2, \dots, L \}$ 의 학습 가능한 분리 평면은 무수히 많이 존재하나, 지지벡터기계는 두 클래스의 분리 경계와 근접한 학습벡터들과 최대 마진(maximal margin) 거리에 위치한 벡터 정보를 이용하여 최종 분리 평면을 결정한다 [1]. 이러한 분리 평면은 최적의 분리 경계를 제공하며 최대 마진 상에 위치한 학습 벡터들을 지지벡터 (support vector)라 한다. 그림 1은 주어진 학습 집합이 선형 분리가 가능한 경우 결정된 분리 평면으로부터 지지벡터까지의 수직거리 γ 은 최대 마진 값이며 모든 지지벡터는 분리 평면으로부터 같은 거리에 위치하게 되는 예이다.

지지벡터기계가 이진 분류 문제의 학습은 최대 마진 상에 위치한 지지벡터들을 찾는 선형 제약식을 갖는 이차 최적화문제(quadratic optimization problem) 식 (4)가된다[1,2]. $C(>0)$ 는 마진과 에러의 학습 모델 복잡도를 조절하는 파라미터이며 슬랙변수 $\xi_i (>0)$ 는 분류 오류를 허용하는 파라미터이다. 식 (4)의 해 (w, b) 로부터 미지의 벡터 x 의 분류함수 (5)는

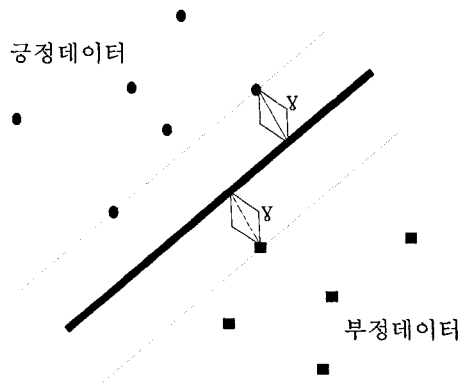


그림 1. 선형 SVM 학습의 예

$h(x|w,b) > 0$ 이면 x 를 긍정 클래스로 $h(x|w,b) < 0$ 인 경우 부정 클래스로 예측한다.

$$\min_{(w,b)} \quad \frac{1}{2} \langle w \cdot w \rangle + C \sum_{i=1}^L \xi_i \quad (4)$$

$$s.t \quad y_i [\langle w \cdot x_i \rangle + b] \geq 1 - \xi_i, i = 1, \dots, L$$

$$h(x|w,b) = \langle w \cdot x \rangle + b \quad (5)$$

식 (4)의 해는 라그랑주(Lagrange) 변수 $a_i, i=1, 2, \dots, L$ 을 도입한 듀얼(dual) 문제 식 (6)으로부터 계산되며 $a_i > 0$ 인 $\langle x_i, y_i \rangle$ 들이 지지벡터의 집합 $SV = \{\langle x_i, y_i \rangle | \langle x_i, y_i \rangle \in S, 0 < a_i < C\}$ 를 구성한다[2].

$$\min_{\alpha} \quad - \sum_{i=1}^L \alpha_i + \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle \quad (6)$$

$$s.t \quad \sum_{i=1}^L y_i \alpha_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, L$$

모델 파라미터 (w,b) 는 라그랑주(Lagrange) 변수와 지지벡터로부터 식 (7)와 같이 계산된다. 테스트 벡터 x 의 클래스는 계산된 파라미터 (w,b) 을 가지고 식(5)로부터 예측된다.

$$w = \sum_{i=1}^L \alpha_i y_i x_i \quad (7)$$

$$b = y_i - \langle w \cdot x_i \rangle, \exists i, 0 < \alpha_i < C$$

비선형 분리 문제에서 선형적 분리 제약식에 따른 최적 해를 얻을 수 없으므로 지지벡터기계는 학습 벡터 x 를 선형 분리가 가능한 고차원 특징 공간(feature space) $\Phi(x)$ 으로 변환시킨 후 두 클래스를 분류하는 선형 분리 평면을 찾는다. 커널함수 $K(x,y)$ 식 (8)를 만족하는 $\Phi(x)$ 의 선택은 지지벡터와 학습 벡터간의 벡터곱(dot product)로 나타나 실질적인 학습 벡터의 고차원 벡터공간으로 매핑은 발생하지 않는다[1,2,4].

$$\Phi: x \in R^n \rightarrow z \in R^m, m \gg n \quad (8)$$

$$K(x,y) = \langle \Phi(x) \cdot \Phi(y) \rangle, x,y \in R^n$$

학습된 분리평면은 고차원 벡터공간에서 선형 분리 경계면이 되며 두 클래스의 지지벡터들과 최대한진 거리상에 위치한다. 테스트 데이터의 예측 오류는 준비된 학습 데이터로부터 얻어진 지지벡터에 의해 의존되어 학습벡터의 차원과는 관계가 없다. 비선형 문제의 해로부터 테스트 벡터를 위한 분류 함수는 지지벡터 집합 SV 와 커널함수로 계산되어 식 (9)가

된다.

$$h(x|w,b) = \sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \quad (9)$$

선택 가능한 커널함수는 선형(linear), 다항식(polynomial), RBF(radial basis function), 시그모이드(sigmoid) 커널 등 다수의 커널이 제시되었다 [1,2,4]. 표 1은 많이 선택되는 커널함수의 예이다. 선형 커널은 선형 분리 가능한 문제에 적용될 수 있으며 비선형 분리 문제에서 커널의 선택은 내재된 데이터들 사이에 관계를 반영할 수 있는 함수로 선택되어야 적절하다. 그러나 준비된 학습 데이터로부터 숨겨진 데이터 구조정보를 이용하는 것은 현실적으로 불가능하여 다수의 실험을 통해 분리 예측율이 높은 커널함수를 선택하게 된다.

3.2 SVM^{light} 기반 다중 분류 학습

지지벡터기계는 이진 분류를 위한 알고리즘이므로 3개 이상의 목적 클래스가 나타나는 다중 분류 문제의 해결은 여러 개의 지지벡터기계를 구성시켜 다중 클래스의 분류로 확장하는 학습전략이 필요하다. 가능한 학습전략은 2절의 관련연구에서 살펴본 학습 기법들이 적용될 수 있다. 다양한 지지벡터기계 알고리즘들이 제시되고 구현되었으며 대표적인 패키지로는 SVM^{light}[2], SMO(Sequential Minimal Optimization[12]), Gist SVM[12], LIBSVM[13] 등이 있다. LIBSVM 패키지는 학습을 쌍별 학습 전략을 이용한 다중 분류 학습 기능을 제공한다. 테스트 결과 SVM^{light}와 LIBSVM이 5,000개 이상의 학습벡터를 갖은 다중 분류 문제에 적용에 덜 민감하였으며 SMO의 구현은 비교적 쉬우나 학습 속도가 SVM^{light}에 비해 늦고 대규모 데이터 집합에서 학습은 가능하지 않았다.

본 논문에서 사용된 단순 이진 분류 학습은 soft-margin 지지벡터기계를 구현한 SVM^{light}의 DLL

표 1. 커널함수의 종류

종류	커널함수	파라미터
선형	$K(x,y) = \langle x \cdot y \rangle + c$	$c \in R$
RBF	$K(x,y) = \exp(-\ x - y\ ^2 / \gamma^2)$	$\gamma \in R$
다항식	$K(x,y) = (s \langle x \cdot y \rangle + c)^d$	$s, c \in R, d \in I$
시그모이드	$K(x,y) = \tanh(\gamma \langle x \cdot y \rangle + c)$	$\gamma, c \in R$

(dynamic linked library)를 생성시켜 사용하였다. Java 환경의 데이터 전처리에서 다중 분류 문제로부터 다수의 이진 분류 문제로 구성하였으며 단일 학습과 쌍별 학습 전략을 채택하여 다중 분류 학습을 위한 지지벡터기계를 구현하였다. SVM^{light}는 C-언어로 구현되어 Java 프로그램에서 학습 및 테스트 모듈을 호출 시 JNI(Java Native Interface)을 이용하였다. 이차 최적화 문제의 해는 PR_LOQO[2,12] 함수를 선택하였다. Windows 사용 환경에서 SVM^{light}의 DLL로부터 관련 함수의 호출은 Java 패키지 xFunction[14]을 사용하였다.

4. 실험 및 분석

지지벡터기계 기반 다중 분류 학습의 성능 평가에 사용된 문제들은 2~26개의 목적 클래스를 포함하고 있다. 학습 벡터의 속성의 수는 4~256 내에 있으며 준비된 데이터의 수는 150~20,000까지 분포한다. 선정된 문제의 속성은 분류, 정수, 실수 데이터로 나타나며 영상처리, 음성인식, 패턴분류 등 다양한 응용 분야에서 선택되었으므로 지지벡터기계 기반 다중 분류 알고리즘의 일반화 성능평가 및 응용성을 살펴볼데 적절하다. 표 2는 성능평가를 위한 다중 분류 문제를 목적클래스 수, 속성 형태, 속성의 수, 학습과 테스트 데이터 수로 기술하고 있다[15-17].

선택된 문제가 학습데이터와 테스트 데이터가 분

리되어 있으면 그대로 사용하였고 그렇지 않은 문제는 목적 클래스에 속한 10%의 데이터를 테스트 데이터로 준비하였다. 데이터 전처리 과정에서 분류 속성의 데이터 값은 자연수로 매핑하였고, 각 속성의 값은 최대값을 가지고 0.0~1.0 사이로 변환시켰으며, Soybean 테스트의 경우 알려지지 않은 값(missing value)은 0으로 바꾸어 실험되었다.

지지벡터기계를 이용한 다중 분류 학습의 성능 평가에서 선택된 커널 함수는 선형, 다항식, RBF 커널 등이다. 각 문제는 선형커널을 가지고 실험하여 학습 데이터의 예측율이 90%이상 이 아니면 다항식 커널과 RBF 커널에 대해 각각 테스트하여 가장 높은 분류 예측율을 보이는 커널을 선택하였다. 커널 파라미터는 선형커널의 $c=0$, 다항식커널의 $d=2, c=0$ 이며 RBF 커널는 $\lambda=1.0$ 으로 선택하였다. 또한 PR_LOQO를 이용한 이차 최적화 문제에서 워킹셋(working set)의 크기는 20으로 설정하였다. 학습 데이터는 10-식 교차검증(10-way cross validation)으로 평가하였으며 이진 분류기의 학습 성능은 학습 데이터에 대해 평균하였고 학습 시간은 초단위로 측정되었다.

신경망을 이용한 실험은 입력-중간-출력의 3층으로 네트워크를 구성하였으며 최소자승(least mean square)의 오류 역전파 학습 알고리즘[3]을 Matlab으로 구현하여 실험하였다. 네트워크의 가중치는 일괄 처리(batch processing)로 학습시켰고 중간층 뉴런의 수는 입력 뉴런의 수의 1.2배로 설정하고, 출력 뉴런의 수는 클래스 수와 같다. 중간층과 출력층 뉴런의 출력 함수는 시그모이드 함수로 선택하였으며, 학습 종료는 허용 오차가 10^{-3} 또는 학습 반복수를 50,000으로 선택하였다. 표 3은 단일 학습과 쌍별 학습 전략 기반 지지벡터기계와 신경망의 다중 분류에 대한 성능 평가 결과이다.

성능 평가 결과가 그림 2과 그림 3에서 비교되었다. 각 학습 모델의 성능 평가에서 가장 우수한 결과는 학습과 테스트로 나누어 볼드체로 제시되었다. 전반적으로 신경망 모델의 학습 결과는 지지벡터기계의 학습 전략과 관계없이 우수하였으나 테스트의 성능은 저조하게 나타났다. 선형커널이 선택된 Iris, Wine, USPS, Soybean 등에서 Wine은 모든 테스트된 분류 학습 모델에서 높은 결과가 나타났으나 전체적으로 학습 성능 평가에서는 신경망 모델이 우수하였다.

표 2. 성능평가를 위한 다중 분류 문제

데이터 집합	속성*	차원	K	학습 데이터 수	테스트 데이터 수
Heart[15]	C,I,R	13	2	191	79
Iris[15]	R	4	3	135	15
Wine[15]	I,R	13	3	178	20
Vehicle[16]	I	18	4	761	87
Glass[15]	R	9	7	192	24
Shuttle[15]	I	9	7	3,866	1,934
USPS[17]	R	256	10	7,291	2,007
Vowel[15]	R	10	11	528	462
Soybean[15]	C	35	19	307	376
Letter[15]	I	16	26	15,000	5,000

* C:분류데이터(categorical data), I:정수(integer), R:실수(real)

표 3. 다중 분류 학습 모델의 성능비교

데이터	단일 학습(OVA)			쌍별 학습(AP)			커널	신경망	
	학습	시간	테스트	학습	시간	테스트		학습	테스트
Heart	84.8	0.5	81.0	84.8	0.5	81.0	RBF	84.9	79.2
Iris	95.6	0.6	93.3	97.0	0.3	100.0	선형	99.6	90.7
Wine	100.0	0.4	100.0	100.0	0.3	100.0	선형	100.0	100.0
Vehicle	98.8	1.8	86.1	99.1	1.2	83.0	RBF	100.0	85.3
Glass	80.1	0.9	74.0	87.4	1.2	69.6	다항식	100.0	54.8
Shuttle	99.7	123.7	99.7	99.8	52.4	99.7	RBF	100.0	99.9
USPS	97.9	1058.7	91.1	100.0	222.3	92.6	선형	99.6	94.5
Vowel	99.6	2.5	52.2	99.8	2.1	60.0	다항식	100.0	50.9
Soybean	96.7	2.1	92.3	96.4	7.5	92.3	선형	100.0	89.3
Letter	99.4	141.9	96.3	99.6	74.1	97.7	RBF	91.6	88.1

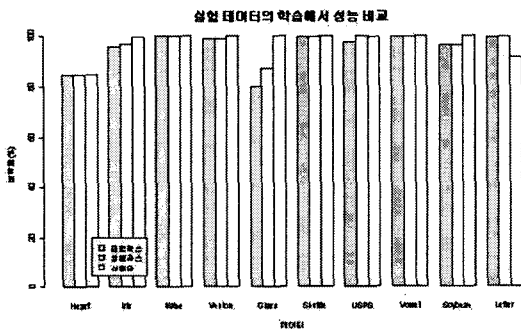


그림 2. 실험데이터의 학습에서 성능 비교

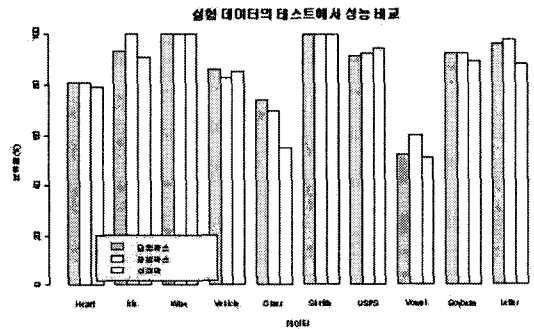


그림 3. 실험 데이터의 테스트에서 성능 비교

RBF 커널의 경우 Shuttle의 실험에서 신경망 모델이 99.9%의 테스트 결과를 보였으나 지지벡터기계의 실험과 미비한 차이를 나타내었다. 같은 커널 함수가 선택된 Heart, Vehicle, Letter 등 실험에서는 신경망 모델이 학습에서 다소 우수한 결과를 보였으나 테스트에서 저조한 결과가 나타났다. 이러한 현상은 다항식 커널 함수를 사용한 Glass와 Vowel 실험에서도 유사하게 나타났다. 지지벡터기계 모델의 선형커널이 선택된 실험에서 단일 학습 전략보다는 쌍별 학습 전략의 성능이 우수하였으며 Soybean 테스트에서는 학습 시간을 제외한 비슷한 결과를 얻었다. RBF 커널의 선택에서는 두 학습전략에 대해 비슷한 성능이 나타났다. 다항식커널이 선택된 Glass, Vowel은 낮은 성능이 나타났지만 선형커널과 RBF의 실험 결과와 비교 시 우수하였으며 단일 학습보다는 쌍별 전략에서 일반화 성능이 높았다. 전반적으로 USPS 문제에서 신경망 모델은 지지벡터기계보

다 비슷하거나 높은 결과를 얻었으나 테스트 과정에서는 지지벡터기계가 낮은 오류율을 나타낸다. 이러한 결과는 신경망 모델의 적응력이 지지벡터기계보다 우수하다는 증거가 된다. 그러나 높은 일반화 성능을 얻는데 높은 학습 적응력이 요구되지 않는다고 하겠다.

전반적으로 쌍별 학습을 하는 지지벡터기계의 학습 시간은 단일 학습보다 시간이 적게 소요되었다. 이전 학습기의 구성에 준비된 데이터의 수가 증가함에 따라 단일 학습 전략이 쌍별 학습 전략에 비해 학습시간이 2배 정도 필요하나 Vehicle, Heart 테스트의 경우에는 이러한 경향은 나타나지 않았다. 신경망 모델의 학습은 지지벡터기계의 학습보다 몇 배의 학습시간이 요구되어 지지벡터기계의 학습 시간과 비교가 되지 않았다. USPS와 Letter 등의 문제에서 신경망 모델은 2~6 시간 정도의 학습시간이 소요되었으나 가장 오랜 학습이 진행된 USPS 문제의 단일

학습 지지벡터기계의 학습 시간은 약 30분이 필요하였다. 학습 데이터의 수와 문제 복잡도가 신경망 모델의 학습시간에 영향을 미치지 않지만 지지벡터기계의 학습에서는 그 영향은 적게 나타났다. RBF 커널의 Shuttle와 Letter, 선형커널의 USPS 등 분류 문제에서는 선택된 커널이 내재된 데이터의 구조를 잘 반영하고 있다는 실험적 근거가 되나 데이터의 수가 학습 시간을 결정하는 주요 요인이 되었다. USPS 문제에서 단일 학습의 학습시간은 쌍별 학습보다 약 4~5배가 더 필요하였다. 이러한 이유는 데이터 불균형 문제로 인해 단일 학습 전략에서는 이진 분류기의 학습에 참여하는 부정 클래스의 데이터의 수가 많아지기 때문으로 분석된다.

5. 결 론

본 논문에서는 다중 분류 문제를 위한 지지벡터기계의 접근 방법을 논의하고 다양한 벤치마킹 분류 문제를 가지고 성능 비교를 수행하였다. 지지벡터기계 알고리즘은 고유한 이진 분류 특성을 수정 없이 분류 문제를 축소하고 조합하는 학습 전략을 이용하면 다중 분류 알고리즘으로 확장이 가능하다. 단일 학습 전략과 쌍별 학습 전략에 따른 지지벡터기계를 다중 분류 문제에 적용하였으며 공개된 데이터를 가지고 실험을 수행하였고, 신경망 모델과 성능을 비교하였다. 실험을 통해 지지벡터기계의 다중 분류 문제에 대한 학습 전략이 효과적이면 일반화 성능의 비교에서 다중 분류 신경망보다 우수하였다. 이진 분류기의 구성은 단일 학습의 경우 학습기의 클래스의 수와 동일하지만 쌍별 학습에서는 $O(K^2)$ 에 비례하는 학습기가 필요하다.

실험 평가에서 단일 학습 전략보다는 쌍별 학습이 학습과 테스트에서 다소 우수한 결과를 얻었다. 신경망 모델과 비교 시 다중 분류를 위한 지지벡터기계가 학습에서 낮은 결과를 얻었으나 테스트에서는 우수하였다. 전반적으로 단일 학습 전략은 쌍별 학습 전략에 비해 학습 시간이 많이 필요했으나 신경망 모델과 비교가 되지 않았다. 단일 학습 전략을 적용 시 이진 분류기의 다중 분류기로의 확장에 따른 데이터 불균형 문제가 발생된다. 데이터 불균형 문제의 해결은 샘플링 기법을 이용한 부정/긍정 클래스에 속한 학습 데이터의 비율을 조정, 학습 데이터의 선택, 또

는 인위적 학습 데이터의 생성 등에 근거한 일반화 성능을 높이는 기법 등이 연구되었다. 이들 연구 결과가 지지벡터기계를 이용한 단일 학습 전략에 내재될 수 있으면 클래스 수와 동일한 이진 분류기의 구성이 가능하여 다중 분류기의 구성 기법의 응용성 및 확장성을 높일 수 있을 것이다.

참 고 문 헌

- [1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: data mining, inference, prediction*, Springer, 2001.
- [2] Thorsten Joachims, *Learning To Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*, Kluwer Academic Publishers, 2001.
- [3] Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press-Oxford, 1995.
- [4] F. van der Heijden, R.P.W. Duin, D. de Ridder, and D.M.J. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach using Matlab*, Wiley, 2004.
- [5] Trevor Hastie and Robert Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, Vol.26, No.3, pp. 451-471, 1998.
- [6] Hsu C-W and Lin C-J, "A comparison of methods for multi-class support vector machines," *IEEE Trans. on Neural Networks*, Vol.13, pp.415-425, 2002.
- [7] Guruswami, V. and Sahai, A., "Multiclass learning, boosting, and error-correcting codes," *In proceeding of the 12th Annual Conference on Computational Learning Theory*, ACM Press, pp. 145-155, 1999.
- [8] Erin L. Allwein, Robert E. Schapire, and Yoram Singer, "Reducing multiclass to binary: a unifying approach for margin classifier," *Journal of Machine Learning Research*, Vol.

1, pp. 113-141, 2000.

[9] Dzeroski S and Zenko B, "Stacking with multi-response model trees," *Proc. of the third international workshop on multiple classifier systems*, MCS, Springer-Verlag, pp. 201-211, 2002.

[10] Tao Li, Shenghuo Zhu and Mitsunori Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and Information Systems*, Vol.10, No.4, pp. 453-472, 2006.

[11] Nathalie Japkowicz, "The Class Imbalance Problem: Significance and Strategies," *In Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, pp. 111-117, 2000.

[12] Kernel Machines, <http://www.kernel-machines.org/software>

[13] Chih-Chung Chang and Chih-Jen Lin, LIBSVM - A Library for Support Vector Machines, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

[14] xFunction, <http://www.excelsior-usa.com/xfunction.html>

[15] UCI Machine Learning Repository, <http://www.ics.uci.edu/MLRepository.html>, University of California, Irvine.

[16] Statlog database, <http://niaad.liacc.up.pt/old-statlog/datasets.html>

[17] Yann LeCun's website, <http://www.research.att.com/yann/exdb/mnist>



황 두 성

1985년 충남대학교 계산통계학과 졸업(학사)
 1990년 충남대학교 대학원 계산통계학과(석사)
 2003년 Wayne State University, Computer Science (박사)

1990년~1991년 국토개발연구원 연구원
 1991년~1998년 전자통신연구소 선임연구원
 2003년~현재 단국대학교 컴퓨터학과
 관심분야 : 데이터 마이닝(data mining), 기계학습(machine learning), 바이오인포매틱스(bioinformatics)