Journal of The Korea Society of Computer and Information
Vol. 19, No. 6, June 2014

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.6.019

Support Vector Machine을 이용한 문맥 인지형 융합

허 경 용*, 김 성 훈**

Context-Aware Fusion with Support Vector Machine

Gyeong-yong Heo*, Seong-Hoon Kim**

요 약

앙상블 분류기는 여러 개의 분류기에서의 예측 결과를 결합함으로써 단일 분류기에 비해 신뢰성 높은 예측 결과를 얻을 수 있는 방법으로 널리 사용되고 있다. 앙상블 분류기를 위해서는 여러 가지 방법이 사용되고 있으며 흔히 사용되는 방법으로는 부스팅이 있다. 하지만 부스팅은 단계적인 학습을 통해 이전 단계에서 잘못 분류된 샘플들을 다음 단계에서 다시 분류하는 방식으로 이전 단계로의 피드백이 불완전한 순차적인 방법이라는 한계가 있다. 이 논문에서는 단일 분류기 중 가장 성능이 좋은 것으로 알려진 SVM을 기본 분류기로 사용하여 동시에 여러 개의 SVM을 학습하는 문맥 감지형 SVM 앙상블 알고리즘을 제안한다. 제안하는 방법에서는 특징 공간을 문맥으로 나누는 클러스터링과 SVM 학습을 동시에 진행하므로 특징 공간 분할과 학습이 서로의 결과를 사용할 수 있어 기존 앙상블 학습에 비해 더 나은 결과를 얻을 수 있으며 이는 실험 결과를 통해 확인할 수 있다.

▸ Keywords : 앙상블, SVM, 문맥, 클러스터링

Abstract

An ensemble classifier system is a widely-used multi-classifier system, which combines the results from each classifier and, as a result, achieves better classification result than any single classifier used. Several methods have been used to build an ensemble classifier including boosting, which is a cascade method where misclassified examples in previous stage are used to boost the performance in current stage. Boosting is, however, a serial method which does not form a complete feedback loop. In this paper, proposed is context sensitive SVM ensemble (CASE) which adopts SVM, one of the best classifiers in term of classification rate, as a basic classifier and clustering method to divide feature space into contexts. As CASE divides feature space and trains

•제1저자 : 허경용 •교신저자 : 김성훈
•투고일 : 2014. 5. 12, 심사일 : 2014. 5. 21, 게재확정일 : 2014. 5. 27.
* 동의대학교 전자공학과(Dept. of Electronic Engineering, Dong-Eui University)
** 경북대학교 컴퓨터정보학부(School of Computer Information, Kyungpook National University)
※ 이 논문은 2013학년도 경북대학교 학술연구비에 의하여 연구되었음.

SVMs simultaneously, the result from one component can be applied to the other and CASE achieves better result than boosting. Experimental results prove the usefulness of the proposed method.

▸ Keywords : Ensemble, SVM, Context, Clustering

I. 서 론

앙상블 학습(ensemble learning)이란 기계 학습의 분류 방법을 통해 여러 개의 분류기를 생성하고 이들의 예측을 결합함으로써 새로운 가설(Hypothesis)을 학습하는 방법으로, 다양한 분류기의 예측 결과를 결합함으로써 단일 분류기보다 신뢰성이 높은 예측값을 얻는 것이 목표로 한다.

대부분의 앙상블 학습에 대한 연구는 결정 트리나 Support Vector Machine(SVM)과 같은 하나의 학습 알고리즘을 사용하는 방법에 대해서 이루어졌다[1, 2]. 이러한 방법들은 학습 알고리즘을 조작하여 다양한 분류기를 생성한 후 다수결(majority voting)이나 가중치 투표(weighted voting)에 의하여 예측값을 결합한다. 이러한 앙상블 학습을 하는 대표적인 방법으로 부스팅(Boosting)과 배깅(Bagging)이 있다[3, 4]. 배깅과 부스팅은 학습 데이터를 샘플링하여 다양한 학습 데이터를 생성하며, 하나의 학습 알고리즘을 적용하여 다양한 분류기를 생성하는 방법이다. 부스팅은 배깅의 단점을 보완하여 이전 모델에서 오분류된 샘플의 선택 확률을 높여 다음 모델에서 오류에 대한 보완을 유도하는 방식으로 AdaBoost가 대표적이다[5]. SVM을 기본 분류기로 사용한 SVM 앙상블에 대한 다양한 연구도 진행되어 왔다[6-8]. 하지만 SVM 앙상블은 SVM을 기본 분류기로 사용하는 부스팅 알고리듬과 다르지 않다.

부스팅 알고리듬은 단순하면서도 강력한 분류기 결합 방법이기는 하지만 순차적 방법이며 분류 오류에 기초한 방법이라는 특징이 있다. 부스팅은 먼저 학습된 분류기의 분류 결과가 다음 분류기의 학습에 정보를 제공하는 구조로 이루어져 있으며, 다음 분류기로 제공되는 정보는 해당 학습 데이터의 분류 결과로 이를 바탕으로 학습 샘플을 선택하게 된다.

이 논문에서는 앙상블 학습 방법의 일종인 문맥 감지형 SVM 앙상블(Context Aware SVM Ensemble, CASE)을 제안하였다. 부스팅이 오분류를 기준으로 다수의 기본 분류기가 학습할 특징 벡터 공간을 나누는 것과 달리 CASE에서는 오분류뿐만이 아니라 특징 벡터의 분포 자체를 기준으로 특징 벡터 공간을 나누다는 점에서 차이가 있다. 특징 벡터 공간의 분할을 위해서는 특징 벡터에 대한 클러스터링이 사용되었다. 또 다른 차이점으로는 부스팅이 K개 기본 분류기를 순차적으로 학습하는 반면 CASE는 K개 분류기를 동시에 학습한다는 점이다. 이처럼 CASE는 균일한 특징 공간의 지역을 나타내는 문맥(context)으로 나누고 각 문맥에서 장점을 가지는 분류기, SVM을 학습시키며 이는 동시에 진행된다. 문맥 학습을 위한 클러스터링은 Fuzzy C-Means(FCM)[9]가 사용되었지만 개발된 다양한 클러스터링을 적용할 수 있으며 단일 목적함수를 가지는 문제로의 형식화가 가능한 장점이 있다. 제안한 CASE를 이용한 실험 결과에 따르면 대표적인 부스팅 알고리듬인 AdaBoost에 비해서도 나은 결과를 보였으며 개선된 문맥 분할 방법이나 변형된 SVM을 적용함으로써 보다 나은 결과를 얻을 수 있을 것으로 기대된다.

이 논문의 구성은 다음과 같다. 2장에서는 CASE의 기본 분류기에 해당하는 SVM 및 퍼지 SVM에 대해 살펴본다. 3장에서는 Fuzzy C-Means와 Fuzzy SVM을 이용하여 CASE를 구성하는 방법을 보이며 4장에서는 실험 결과를 통해 제안한 방법의 실효성을 보인다. 결론 및 향후 연구 방향에 대해서는 5장에서 제시한다.

II. SVM과 퍼지 SVM

1. SVM (Support Vector Machine)

Support Vector Machine(SVM)은 통계적 학습 이론에 근거한 교사 학습 방법의 일종으로 기존의 통계적 학습 방법들이 경험적 위험 최소화 (empirical risk minimization) 방법을 사용하는 반면 구조적 위험 최소화 (structural risk minimization) 방법을 통해 일반화 오류를 감소시키는 방법을 채택하고 있다[10]. SVM은 1970년대 후반 Vapnik에 의해 제안되었으며 1990년대 필기 인식[10], 문서 분류[11], 이미지 검색[12] 등 다양한 분야에서 성공적인 결과를 보여줌으로써 주목을 받게 되었다.

학습 데이터 $\{(x_i,y_i)|x_i \in R^d, y_i \in R, 1 \leq i \leq N\}$이 주어졌다고 가정하자. 이 때 x_i는 d 차원 특징 벡터를 나타내며, y_i는 클래스 라벨로 {-1, 1}의 값을 가지는 것이 일반적이다. SVM은 일반적으로 두 개의 클래스로 나누는 문제(two class problem)를 다루며, 이를 일반화한 세 개 이상의 클래스로 나누는 문제(multi class problem)에도 적용이 가능하지만 이 논문에서는 전통적인 SVM에서 다루는 두 개의 클래스로 나누는 문제로 한정한다. 데이터가 주어진 경우 SVM은 두 클래스를 나누면서 분리 간격을 최대로 하는 경계면을 찾아내는 것을 그 목적으로 하며 경계면은 식 (1)의 제약조건을 만족시켜야 한다.

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i = 1,...,N \tag{1}$$

이 때 w와 b는 경계면을 나타내는 벡터와 절편을, $\xi_i \geq 0$는 오류값을 나타낸다. 식 (1)의 식을 만족시키는 최적의 분리 경계면을 찾는 문제는 식 (2)의 최적화 문제로 표현될 수 있다.

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i = 1,...,N \\ & \xi_i \geq 0 \quad \forall i = 1,...,N \end{aligned} \tag{2}$$

이 때 C는 SVM의 정규화(regularization) 상수를 나타낸다. 식 (2)의 최적화 문제는 라그랑지 방정식을 이용하여 식 (3)의 dual problem으로 간략화할 수 있다.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \forall i = 1,...,N \end{aligned} \tag{3}$$

이 때 $\alpha = (\alpha_1,...,\alpha_N)$은 라그랑지 상수를 나타낸다. 식 (3)은 식 (2)와 달리 에러값을 나타내는 ξ_i가 없으며 이차 프로그래밍 (quadratic programming) 최적화 기법을 통해 해를 구할 수 있다. 식 (3)의 해는 식 (4)와 (5)의 제약조건을 만족시켜야 한다.

$$\alpha_i(y_i(w \cdot x_i + b) - 1 + \xi_i) = 0 \quad \forall i = 1,...,N \tag{4}$$

$$(C - \alpha_i)\xi_i = 0 \quad \forall i = 1,...,N \tag{5}$$

$\alpha_i \neq 0$인 데이터 포인트들은 support vector(SV)라 불리며 경계면 형성에 영향을 미친다. SV는 두 가지 종류가 있으며 $0 < \alpha_i < C$인 SV는 식 (5)에 의해 $\xi_i = 0$을 만족하며 식 (4)에 의해 $y_i(w \cdot x_i + b) = 1$을 만족한다. $\alpha_i = C$인 SV는 ξ_i가 0이 아니며 구해진 분류면에 의해 분류를 시행할 경우 잘못 분류되는 데이터 포인트들에 해당한다. $\alpha_i = 0$인 데이터 포인트들은 분류 경계면에서 멀리 떨어져 있어 경계면 형성에 영향을 미치지 않으며 분류 시 에러도 발생하지 않는다. 최적의 분류 경계면은 식 (3)에서 얻어지는 α를 이용하여 식 (6)과 같이 얻어지는 벡터와 식 (4)에서 얻어지는 절편 b를 통해 구할 수 있다.

$$w = \sum_{i=1}^{N}\alpha_i y_i x_i \tag{6}$$

새로운 데이터 포인트 x가 주어진 경우의 판별식은 식 (7)과 같다.

$$f(x) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i (x_i \cdot x) + b\right) \tag{7}$$

2. Fuzzy SVM

SVM의 성공에도 불구하고 여전히 SVM의 잡음 민감성의 문제를 가지고 있다. 잡음 민감성 문제는 모든 분류기들이 공통으로 가지는 문제이며 SVM의 경우 오류값 ξ_i와 정규화 상수 C를 통해 다른 분류기에 비해 잡음 민감성 문제가 덜하지만 완전히 자유로울 수는 없다. 잡음 민감성의 문제를 완화하기 위해서는 다양한 방법이 사용되며 그 중 하나가 퍼지 소속도를 이용한 퍼지 분류기로의 변형이다. SVM의 경우에도 fuzzy SVM (FSVM)[13]을 비롯하여 new fuzzy SVM[14], prior knowledge SVM[15], posterior probability SVM[16], soft SVM[17] 등이 퍼지 소속도를 이용하여 잡음 민감성의 문제를 해결하고자 하였다. 이들은

목적 함수에서 약간의 차이가 있지만 데이터 포인트 별로 서로 다른 가중치를 적용하고자 하였다는 점에서 동일하다. 보다 근본적인 문제는 데이터 포인트에 적용할 가중치를 어떻게 적용할 것인가 하는 문제이다.

$\{(x_i, y_i, u_i) | x_i \in R^d, y_i \in R, 0 \leq u_i \leq 1, 1 \leq i \leq N\}$ 의 학습 데이터가 주어졌다고 가정하자. SVM에서와는 다르게 각 데이터 포인트에는 퍼지 소속도가 주어지며 이 값은 SVM 적용 이전에 결정되어야 한다. 분리 간격을 최대로 하는 최적의 분리면을 찾는 문제는 식 (8)과 같이 나타낼 수 있으며 이를 Fuzzy SVM(FSVM)이라 한다.

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} u_i \xi_i \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i = 1,...,N \\ & \xi_i \geq 0 \quad \forall i = 1,...,N \end{aligned} \tag{8}$$

식 (8)은 식 (2)와 달리 각 데이터 포인트의 오류값이 소속도의 영향을 받는다. 즉, 소속도가 작은 데이터 포인트는 큰 오류값을 가질 수 있다. 소속도가 작은 값은 일반적으로 잡음이 많이 포함된 데이터이므로 이들 데이터가 목적함수의 최적화에 미치는 영향을 줄임으로써 잡음 민감성을 줄일 수 있다.

식 (8)은 라그랑지 방정식을 이용하여 식 (9)의 dual problem으로 변환할 수 있다.

$$\begin{aligned} \max \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq u_i C \quad \forall i = 1,...,N \end{aligned} \tag{9}$$

식 (9)의 해는 식 (10)과 (11)의 제약조건을 만족시켜야 하며 식 (10)은 식 (4)와 동일함을 알 수 있다.

$$\alpha_i(y_i(w \cdot x_i + b) - 1 + \xi_i) = 0 \quad \forall i = 1,...,N \tag{10}$$

$$(u_i C - \alpha_i)\xi_i = 0 \quad \forall i = 1,...,N \tag{11}$$

FSVM에서도 $\alpha_i \neq 0$인 데이터 포인트들은 support vector(SV)라 불리며 경계면 형성에 영향을 미친다. SV는 두 가지 종류가 있으며 $0 < \alpha_i < u_i C$인 SV는 식 (11)에 의해 $\xi_i = 0$을 만족하며 식 (10)에 의해 $y_i(w \cdot x_i + b) = 1$을 만족한다. $\alpha_i = u_i C$인 SV는 잘못 분류되는 데이터를 나타낸다. $\alpha_i = 0$인 데이터 포인트들은 분류 경계면에서 멀리 떨어져 있는 점들이다. 식 (10)과 (11)을 SVM과 비교했을 때 소속도 u_i가 포함되어 있다는 점을 제외하면 기본적으로 동일하다.

식 (9)의 해인 α값은 소속도 값을 반영하여 결정되므로 새로운 데이터 포인트 x가 주어진 경우의 판별식은 식 (7)과 동일하다.

III. 문맥 감지형 SVM 앙상블

이 장에서는 특징 공간을 주어진 개수만큼의 균일한 부분집합으로 나누고 각 부분집합의 특성에 맞는 인식기를 할당함으로써 인식기의 성능을 높일 수 있는 방법을 제시한다. 그림 1은 제안하는 문맥 감지형 SVM 앙상블 (CASE, Context Aware SVM Ensemble)의 개념을 나타낸 것이다.

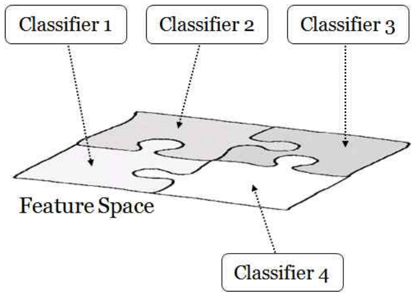


그림 1. CASE의 개념도
Fig. 1. Conceptual diagram of CASE

그림 1에서 알 수 있듯이 문제의 특징 공간은 그 분포에 따라 균일한 부분집합으로 나뉘어져 있으며 각 부분집합에는 전용의 분류기가 할당되어 있다. 이 때 특징 공간의 균일한 부분집합을 문맥(context)이라 하며 각 문맥에서 동작하는 인식기는 SVM을 사용한다. 균일한 문맥을 찾아내기 위해 CASE에서는 클러스터링을 이용한다. 클러스터링은 데이터가 주어지는 경우 주어진 척도에서 유사한 데이터들끼리 군집을 형성하도록 하는 기법이므로 CASE에서 균일한 문맥을 찾아내기 위해 사용할 수 있다. 따라서 CASE의 목적 함수는 균일한 문맥을 찾아내기 위한 클러스터링 목적함수와 문맥에 맞는 분류기 최적화를 위한 SVM 목적함수의 조합으로 구성할 수 있다.

N개의 특징 벡터 $X = \{x_i | x_i \in R^d, i = 1,...,N\}$와 기대

출력 $Y=\{y_i|y_i\in\{-1,1\},i=1,...,N\}$가 주어진 경우 K 개의 클러스터 구성을 위한 목적 함수는 식 (12)와 같다.

$$J_1=\sum_{i=1}^{N}\sum_{k=1}^{K}I(x_i,k)\|x_i-v_k\|^2 \tag{12}$$

이 때 $I(x_i,k)$는 x_i가 k번째 클러스터에 포함될 경우 1을 아닌 경우에는 0을 반환하는 지시함수(indicator function)이며, v_k는 k번째 클러스터의 중심을 나타낸다. CASE에서는 클러스터 개수만큼의 SVM이 존재하므로 식 (13)의 목적함수를 최소화하여야 한다.

$$J_2=\sum_{k=1}^{K}\left(\frac{1}{2}\|w_k\|^2+C\sum_{i=1}^{N}I(x_i,k)\xi_{ki}\right) \tag{13}$$

식 (13)에서 w_k는 k번째 SVM의 분리면을 나타내며 ξ_{ki}는 k번째 SVM에서 x_i의 오류를, C는 k개 SVM이 공유하는 정규화 상수를 나타낸다. 식 (13)에서도 지시함수가 사용되었으며 이는 그림 1에 나타난 바와 같이 각 클러스터에 속하는 입력에 대하여 SVM이 지역적으로 최적화를 시도하기 때문이다.

CASE의 목적함수는 식 (12)와 (13)의 조합을 최소화시키는 것으로 정의할 수 있으며 식 (14)와 같이 나타낼 수 있다.

$$\begin{aligned}J=&\sum_{i=1}^{N}\sum_{k=1}^{K}I(x_i,k)\|x_i-v_k\|^2\\&+\sum_{k=1}^{K}\left(\frac{1}{2}\|w_k\|^2+C\sum_{i=1}^{N}I(x_i,k)\xi_{ki}\right)\end{aligned} \tag{14}$$

식 (14)의 목적함수에서 지시함수 $I(\cdot)\in\{0,1\}$을 퍼지 소속도 $u_{ki}\in[0\ 1]$로 교체하면 식 (14)를 포함하는 보다 일반적인 식을 얻을 수 있다. 이 경우 J_1은 HCM(Hard C-Means)의 목적함수에서 FCM(Fuzzy C-Means)의 목적함수로, J_2는 SVM의 목적함수에서 FSVM의 목적함수로 바뀌게 된다. 퍼지 소속도를 도입한 CASE의 목적함수는 식 (15)와 같이 나타낼 수 있다.

$$\begin{aligned}J_{CASE}=&\sum_{i=1}^{N}\sum_{k=1}^{K}u_{ki}^{m}\|x_i-v_k\|^2\\&+\alpha\sum_{k=1}^{K}\left(\frac{1}{2}\|w_k\|^2+C\sum_{i=1}^{N}u_{ki}^{m}\xi_{ki}\right)\end{aligned} \tag{15}$$

이 때 m은 퍼지화 상수(fuzzifier constant)를 나타내며 α는 클러스터링 항과 SVM 항의 비를 나타내는 상수이다. 식 (15)의 해는 해석적인 방법으로 구할 수 없으며 반복 최적화(iterative optimization)을 통해 구하여야 한다. 식 (15)에서 소속도는 식 (16)의 제약조건을 만족시켜야 한다.

$$\sum_{k=1}^{K}u_{ki}=1 \tag{16}$$

식 (15)와 (16)을 이용하여 식 (17)과 같이 라그랑지 방정식을 구성할 수 있다.

$$\begin{aligned}\mathcal{L}=&\sum_{i=1}^{N}\sum_{k=1}^{K}u_{ki}^{m}\|x_i-v_k\|^2\\&+\alpha\sum_{k=1}^{K}\left(\frac{1}{2}\|w_k\|^2+C\sum_{i=1}^{N}u_{ki}^{m}\xi_{ki}\right)\\&-\sum_{i=1}^{n}\lambda_i\left(\sum_{k=1}^{K}u_{ki}-1\right)\end{aligned} \tag{17}$$

먼저 식 (17)를 u_{ki}에 대하여 미분하면 식 (18)를 얻을 수 있다.

$$\frac{\partial\mathcal{L}}{\partial u_{ki}}=mu_{ki}^{m-1}d_{ki}^2+\alpha Cmu_{ki}^{m-1}\xi_{ki}-\lambda_i \tag{18}$$

이 때 $d_{ki}=\|x_i-v_k\|$를 나타낸다. 식 (18)을 영(zero)으로 놓고 정리하면 식 (19)와 같은 u_{ki}의 갱신식을 얻을 수 있다.

$$u_{ki}=\frac{(\lambda_i/m)^{\frac{1}{m-1}}}{\left(\overline{D}_{ki}\right)^{\frac{1}{m-1}}} \tag{19}$$

이 때 $\overline{D}_{ki}$는 식 (20)와 같이 정의된다.

$$\overline{D}_{ki} = \|x_i - v_k\|^2 + \alpha C \xi_{ki} \tag{20}$$

식 (20)의 거리(distance)는 FCM에서 사용되는 클러스터 중심에서의 거리인 비교사학습 거리뿐만이 아니라 SVM에서 분류 결과로 생성되는 교사학습 거리인 ξ를 포함하고 있어 소속도 계산과 분류기 학습이 상호작용하도록 해준다. 식 (19)는 식 (16)의 제약조건을 만족시켜야 하므로 식 (19)를 식 (16)에 대입하면 식 (21)을 얻을 수 있다.

$$\sum_{k=1}^{K} u_{ki} = \sum_{k=1}^{K} \frac{(\lambda_i/m)^{\frac{1}{m-1}}}{\left(\overline{D}_{ki}\right)^{\frac{1}{m-1}}} = 1 \tag{21}$$

$$(\lambda_i/m)^{\frac{1}{m-1}} = \frac{1}{\sum_{k=1}^{K}\left(\frac{1}{\overline{D}_{ki}}\right)^{\frac{1}{m-1}}}$$

식 (19)와 (21)을 이용하여 소속도 u_{ki}에 대한 갱신식은 식 (22)와 같이 나타낼 수 있다.

$$u_{ki} = \frac{\left(\frac{1}{\overline{D}_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{a=1}^{K}\left(\frac{1}{\overline{D}_{ai}}\right)^{\frac{1}{m-1}}} \tag{22}$$

식 (17)를 v_k에 대하여 미분하면 식 (23)를 얻을 수 있다.

$$\frac{\partial\, \mathcal{L}}{\partial\, v_k} = -2\sum_{i=1}^{N} u_{ki}^m (x_i - v_k) \tag{23}$$

식 (23)을 영으로 놓고 정리하면 클러스터 중심 v_k에 대한 갱신식 식 (24)를 얻을 수 있다.

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m x_i}{\sum_{i=1}^{N} u_{ki}^m} \tag{24}$$

식 (15)에는 u_{ki}와 v_k 이외에도 k개 SVM의 최적 분류 경계면 w_k와 각 SVM에서의 오류값 ξ를 최적화시켜야 하며 이는 식 (9)의 dual problem 최적화를 통해 얻어낼 수 있다. 식 (22) 및 (24)의 갱신식과 FSVM의 최적화를 이용한 CASE 학습 알고리듬은 그림 2와 같이 나타낼 수 있다.

1. 소속도 U 초기화
2. repeat
3. 클러스터 중심 V 갱신. 식 (24)
4. 소속도 U 갱신. 식 (22)
5. SVM 파라미터 갱신
6. until 수렴

그림 2. CASE 알고리듬
Fig. 2. CASE algorithm

IV. 실험 결과

제안하는 알고리듬의 유용성을 보이기 위해 CASE와 AdaBoost를 Matlab으로 구현하여 비교하였다. 실험에 사용한 데이터는 인공 데이터로, 두 개의 클래스로 구성되고 원형으로 배치된 데이터를 사용하였다. AdaBoost는 선형 분리기를 사용하고 사용한 분류기의 개수에 의해 성능이 향상된다. CASE의 경우 SVM을 사용하므로 AdaBoost와 동일한 비교를 위해 선형 커널(linear kernel)을 사용하였다. 그림 3은 실험에 사용한 데이터의 예이다.

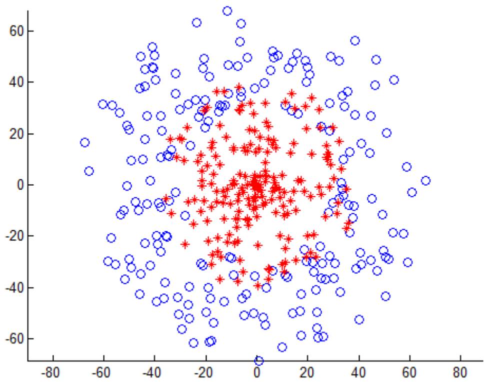


그림 3. 데이터 집합
Fig. 3. Data set

그림 4는 AdaBoost와 CASE에서 분류기의 개수가 증가함에 따라 오류가 감소하는 예를 보여고 있다. CASE의 경우 분류기 개수의 증가는 문맥의 증가, 즉, 클러스터 개수의 증가를 의미한다. AdaBoost의 경우 현재 단계에서 잘못 분류

된 데이터가 다음 단계에서의 주 분류 대상이 되므로 단계 사이의 피드백이 완전하지 않아 분류기의 개수가 증가하는 경우에도 오류는 증가하는 경우가 발생할 수 있으며, 잡음에 민감한 단점이 있다. 반면 CASE의 경우에는 문맥 분리와 분류기 학습이 동시에 진행되므로 분류기의 개수가 증가하는 경우 오류가 감소한다. 또한 소속도의 도입과 SVM에서의 정규화 상수 C로 인해 잡음 영향을 적게 받는 것을 알 수 있다.

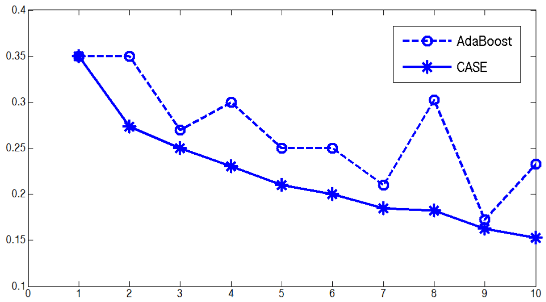


그림 4. AdaBoost와 CASE 비교
Fig. 4. Comparision between AdaBoost and CASE

AdaBoost와의 비교를 위해 CASE에서 경우 선형 커널을 사용하였으므로 10개의 클러스터까지 필요하였지만, 비선형 커널을 사용하는 경우에는 보다 적은 개수의 클러스터에서도 우수한 성능을 보여줄 것으로 예상된다.

V. 결론 및 보완점

이 논문에서는 여러 개의 분류기의 결과를 종합하여 보다 신뢰성이 높은 결과를 얻을 수 있는 새로운 앙상블 분류기인 문맥 민감형 SVM 앙상블(CASE)을 제안하였다. 제안한 알고리듬은 주어진 특징 공간을 균일한 여러 개의 문맥으로 분리하고 분리된 문맥에서의 분류기 학습을 동시에 진행함으로써 잡음에 강하고 안정적인 분류기 학습이 가능하다. 이러한 사실은 AdaBoost와 비교 실험을 통해 확인할 수 있다.

CASE가 AdaBoost에 비해 보다 안정적이고 잡음에 강한 결과를 보여주고는 있지만 AdaBoost에 비해 SVM을 사용하는 한계로 인해 학습 시간이 많이 걸리는 단점이 있다. 이는 비선형 커널을 도입하여 클러스터의 개수를 줄임으로써 해결할 수 있을 것으로 예상되지만, 클러스터 개수를 결정하는 문제는 여전히 풀어야할 문제로 남아있으며 현재 이에 관해 연구 중에 있다.

성능 향상을 위해 고려할 수 있는 다른 방법 중 하나는 클러스터링 방법의 변경이 있다. 현재 CASE에서는 기본적인 퍼지 클러스터링을 사용하지만 이는 비교사 학습 방법으로 분류 측면에서 최적임을 보장하지는 못한다. 따라서 교사 클러스터링을 도입함으로써 분류 측면에서 최적의 결과를 얻을 수 있을 것으로 생각되며 교사 클러스터링과 CASE의 결합에 대해서도 연구 중에 있다.

참고문헌

[1] D.T. G. Dietterich, “Ensemble Methods in machine learning,” Proceedings of the First International Workshop on Multiple Classifier Systems, pp. 1-15, 2000.

[2] L. Rokach, ″Ensemble-based classifiers,″ Artificial Intelligence Review, Vol. 33, No. 1-2, pp. 1-39, Feb. 2010.

[3] L. Breiman, “Bagging predictors,” Machine Learning, Vol. 24, No. 2, pp. 123-140, Aug. 1996.

[4] Boosting.org, *http://www.boosting.org*

[5] Robert E.Schapire ″A decision-theoretic generalization of on-line learning and an application to boosting,″ Journal of Computer and System Sciences, Vol. 55, No. 1, pp. 119-139, Aug. 1997.

[6] Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim and Sung Yang Bang, “Constructing support vector machine ensemble,” Pattern Recognition, Vol. 36, No. 12, pp. 2757-2767, Dec. 2003.

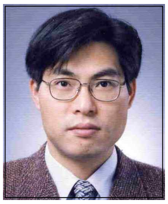
[7] Zhi-Yong Lin, Zhi-Feng Hao, Xiao-Wei Yang and XiaoLan Liu, “Several SVM Ensemble Methods Integrated with Under-Sampling for Imbalanced Data Learning,” Advanced Data Mining and Applications, Lecture Notes in Computer Science, Vol. 5678, pp. 536-544, 2009.

[8] E. Ahmed, N. El-Gayar and I. A. El-Azab, “Support Vector Machine ensembles using features distribution among subsets for enhancing microarray data classification,” Proceedings of the 10th International Conference on Intelligent Systems Design and Applications, pp. 1242-1246, 2010.

[9] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.
[10] V. Vapnik, "Statistical Learning Theory," John Wiley & Sons, New York, 1998.
[11] K. Wu and K. H. Yap, "Fuzzy SVM for content-based image retrieval," IEEE Computational Intelligence Magazine, Vol. 1, No. 2, pp. 10-16, May 2006.
[12] B. Lim, M. Tsui, V. Charastrakul and D. Shi, "Web search with text categorization using probabilistic framework of SVM," Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 2950-2955, 2006.
[13] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," IEEE Transactions on Neural Networks, Vol. 13, No. 2, pp. 464-471, March 2002.
[14] Y. Wang, S. Wang and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," IEEE Transactions on Fuzzy Systems, Vol. 13, No. 6, pp. 820-831, Dec. 2005.
[15] L. Wang, P. Xue and K. L. Chan, "Incorporating prior knowledge into SVM for image retrieval," Proceedings of the 17th International Conference on Pattern Recognition, pp. 981-984, 2004.
[16] Q. Tao, G. W. Wu, F. Y. Wang and J. Wang, "Posterior probability support vector machines for unbalanced data," IEEE Transactions on Neural Networks, Vol. 16, No. 6, pp. 1561-1573, Nov. 2005.
[17] Y. Liu and Y. F. Zheng, "Soft SVM and its application in video-object extraction," IEEE Transactions on Signal Processing, Vol. 55, No. 7, pp. 3272-3282, July 2007.

저 자 소 개

허 경 용
1994: 연세대학교
전자공학과 공학사.
1996: 연세대학교 대학원
전자공학과 공학석사.
2009: University of Florida
컴퓨터공학과 공학박사
현 재: 동의대학교 전자공학과 교수
관심분야: 인공지능, 패턴인식, 로봇공학
Email : hgycap@deu.ac.kr



김 성 훈
1988: 서강대학교
전자공학과 공학사.
1990: 연세대학교 대학원
전자공학과 공학석사.
1996: 연세대학교 대학원
전자공학과 공학박사
현 재: 경북대학교
컴퓨터정보학부 교수
관심분야: 인공지능, 패턴인식
Email : shkim1454@knu.ac.kr