패턴 분류를 위한 Fuzzy Twin Support Vector machine 개발

Development of Fuzzy Support Vector Machine for Pattern Classification

천민규[1], 윤창용[1], 김은태[1], 박민용[1]

[1] 서울시 서대문구 연세대학교 전기전자공학과
E-mail: 1000minkyu@yonsei.ac.kr

요 약

Support Vector Machine(SVM)은 통계적 학습 이론에 기반을 둔 분류기이다. 또한 Twin Support Vecrtor Machine(TWSVM)은 이진 SVM 분류기의 한 종류로써, 서로 관련된 두 개의 SVM 유형 문제를 통해 평행하지 않은 두 개의 평면을 결정하고 이 두 평면을 통해 분류기를 완성하는 방식이다. 이러한 방식의 TWSVM은 학습 시간이 SVM에 비해 훨씬 짧으며, SVM과 비교하여 떨어지지 않는 성능을 보여준다. 본 논문은 분류기 입력에 Fuzzy Membership을 적용하는 방식의 TWSVM을 제안하고, 2차원 벡터 입력에 대한 실험을 통하여 기존에 제시 되었던 TWSVM과 비교한다.

Key Words : 패턴 분류, Support Vector Machine, Twin Support Vector Machine, Fuzzy Membership

1. 서 론

Support Vector Machine(SVM)은 패턴을 고차원 특징 공간으로 사상시킬 수 있다는 점과 대역적으로 최적의 식별이 가능하다는 특징을 가진다. 여기서 가중치는 일반적인 신경망과는 달리 선형 부등 조건을 가진 Quadratic Programming (2차 계획법) 문제를 해결함으로써 얻어지게 된다. 또한 전통적인 대부분의 패턴인식 기법들은 학습 데이터의 수행도를 최적화하기 위한 경험적인 위험 최소화 방법에 기초하고 있는데 반하여, SVM은 고정되어 있지만 알려지지 않은 확률 분포를 갖는 데이터에 대해 잘못 분류하는 확률을 최소화하는 구조적인 위험 최소화 방법에 기초하고 있다. [5]

최근에 제시된 Twin Support Vector Machine (TWSVM)은 분류된 클래스 마다 Quadratic Programming 문제를 적용, 해결하여 클래스마다 서로 평행하지 않는 경계들을 생성한 뒤, 각각의 경계를 이용하여 새로운 분류기를 형성하는 방식이다.

본 논문에서는 최근에 제시된 Twin Support Vector Machine에 Fuzzy Membership을 적용하여, 기존의 Twin Support Vector Machine과의 성능을 비교한다.

2. Twin Support Vector Machine

Twin Support Vector Machine은 군집화된 각 클래스의 데이터 주위에 평행하지 않은 경계를 형성한다. Twin Support Vector Machine의 각 클래스에 대한 두 개의 2차 계획법 문제는 전형적인 Support Vector Machine의 2차 계획법 문제와 형태가 일치한다. [1]

TWSVM 분류기는 다음의 두 개의 2차 계획법 문제를 해결함으로써 얻어지게 된다.

(TWSVM 1)

$$\min_{w^{(1)},\, b^{(1)},\, q} \ (Aw^{(1)}+e_1b^{(1)})^T(Aw^{(1)}+e_1b^{(1)})+c_1e_2^Tq$$
$$-(Bw^{(1)}+e_2b^{(1)})+q \ge e_2, \quad q \ge 0 \qquad (1)$$

(TWSVM 2)

$$\min_{w^{(2)},\, b^{(2)},\, q} \ (Bw^{(2)}+e_2b^{(2)})^T(Aw^{(2)}+e_2b^{(2)})+c_2e_1^Tq$$
$$-(Aw^{(2)}+e_1b^{(2)})+q \ge e_1, \quad q \ge 0 \qquad (2)$$

이 때, A와 B 행렬은 각각 +1과 -1 클래스에 속하는 데이터 행을 포함한 행렬이다. 클래스 +1에 속하는 데이터의 수가 m_1, 클래스 -1에

속하는 데이터의 수가 m_2, $x \in R^n$일 경우, A 행렬은 $m_1 \times n$, B 행렬은 $m_2 \times n$이 된다. c_1, c_2는 모두 0보다 크며, e_1과 e_2 벡터는 행렬 연산에 있어 차원을 맞추어 주기 위한 모든 요소가 1인 벡터이다. 벡터 q는 잘못 분류된 학습 데이터에 해당하는 음이 아닌 에러 변수이다.

이 알고리즘은 각각의 클래스에 속한 데이터들로 부터 거리의 합이 가장 작은 평면을 찾는 과정이다.

식(1), (2)의 첫 번째 수식은 목적 함수로써 하나의 클래스의 데이터들과 평면까지의 거리의 합과 에러 변수의 합을 최소화하는 과정을 의미한다. 이 때, c_1과 c_2는 평면과 데이터 사이 거리 합의 최소화와 에러 변수 합의 최소화의 중요도를 결정하는 변수이다. 즉, c가 크다면 평면과 데이터 사이 거리의 최소화 보다 잘못 분류된 학습 데이터에 해당하는 에러 변수를 최소화가 우선시 될 것 이며, c가 작다면 에러 변수 최소화 보다는 평면과 데이터 간의 거리 최소화가 중요시되는 알고리즘이 될 것이다. 두 번째 수식은 제약식으로써 다른 클래스에 속한 데이터로부터 평면까지의 거리가 적어도 1만큼 떨어져 있어야 함을 의미한다.

다시 말해, TWSVM은 각 클래스에 대한 목적 함수와 다른 클래스 패턴에 의해 결정되는 제약식을 결정하는 2차 계획법 문제를 포함한다. 즉, TWSVM1은 클래스 +1의 데이터들을 $x^T w^{(1)} + b^{(1)} = 0$주위로 군집화하고, TWSVM2는 클래스 -1의 데이터들을 $x^T w^{(2)} + b^{(2)} = 0$로 군집화 한다.

그리고, 데이터의 수가 m이라 가정하면, 일반적인 SVM의 복잡도는 m^3이 되고, TWSVM의 경우는 이를 두 종류의 문제로 각각 나누어 복잡도를 측정하게 된다. 가령 각각의 데이터 수를 $m/2$라고 한다면 SVM과 TWSVM의 실행 시간 비율은 다음과 같다.

$$\left[\left(m^3 \Big/ \left(2 \times \left(\frac{m}{2} \right)^3 \right) \right) \right] = 4$$

즉, TWSVM의 학습 속도가 일반적인 SVM에 비해 4배 정도 빠름을 알 수 있다.

식 TWSVM1 (1)을 통해 라그랑지안 함수를 유도하면 다음과 같다.

$$\begin{aligned} L(w^{(1)}, b^{(1)}, q, \alpha, \beta) = & \frac{1}{2}(Aw^{(1)} + e_1 b^{(1)})^T (Aw^{(1)} + e_1 b^{(1)}) \\ & + c_1 e_2^T q - \alpha^T (-(Bw^{(1)} + e_2 b^{(1)}) + q - e_2)) - \beta^T q \end{aligned} \tag{3}$$

이 때, α와 β는 라그랑제 승수로써 각각 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_{m_2})^T$, $\beta = (\beta_1, \beta_2, \cdots, \beta_{m_2})^T$이다. 그리고 위의 라그랑지안 함수 식 (3)에 대한 K.K.T. (Karush-Kuhn-Tucker)조건을 정리하면 다음과 같다.

$$A^T (Aw^{(1)} + e_1 b^{(1)}) + B^T \alpha = 0 \tag{4}$$

$$e_1^T (Aw^{(1)} + e_1 b^{(1)}) + e_2^T \alpha = 0 \tag{5}$$

$$c_1 e_2 - \alpha - \beta = 0 \tag{6}$$

$$-(Bw^{(1)} + e_2 b^{(1)}) + q \geq e_2,\ q \geq 0 \tag{7}$$

$$\alpha^T (-(Bw^{(1)} + e_2 b^{(1)}) + q - e_2) = 0,\ \beta^T q = 0 \tag{8}$$

$$\alpha \geq 0,\ \beta \geq 0 \tag{9}$$

이 때, $\beta \geq 0$ 이므로, (6)을 통해

$$0 \leq \alpha \leq c_1 \tag{10}$$

를 알 수 있고, 식(4)와 (5)를 함께 정리하면 다음과 같다.

$$[A^T \ \ e_1^T][A \ \ e_1][w^{(1)}, b^{(1)}]^T + [B^T \ \ e_2^T]\alpha = 0 \tag{11}$$

$H = [A \ \ e_1]$, $G = [B \ \ e_2]$, $u = [w^{(1)}, b^{(1)}]$라 하면 식 (11)은 다음과 같이 정리될 수 있다.

$$H^T H u + G^T \alpha = 0,\ 즉,\ u = -(H^T H)^{-1} G^T \alpha \tag{12}$$

식 (12)에서의 벡터 u는 경계면을 결정하는 벡터인 $w^{(1)}$과 변수 $b^{(1)}$을 포함하고 있다. 또한 벡터 u를 구하기 위한 α는 라그랑지안 함수인 식 (3)과 K.K.T 조건을 결합한 아래의 최적화 문제를 해결함으로써 얻을 수 있다.

$$\underset{\alpha}{Max} \ \ e_2^T \alpha - \frac{1}{2} \alpha^T (G(H^T H)^{-1} G) \alpha \tag{13}$$
$$0 \leq \alpha \leq c_1$$

위와 같은 방법으로 TWSVM2에 대해서 정리하면 다음과 같다.

$$\underset{\alpha}{Max} \ \ e_1^T \gamma - \frac{1}{2} \gamma^T (H(G^T G)^{-1} H) \gamma \tag{14}$$
$$0 \leq \gamma \leq c_2$$

또한, TWSVM2의 $w^{(2)}$와 $b^{(2)}$를 포함하는 벡터 $v = [w^{(2)}, b^{(2)}]$는 다음과 같이 나타낼 수 있다.

$$v = (G^T G)^{-1} H^T \gamma \tag{15}$$

식 (12)와 (15)를 통해 클래스 +1과 클래스 -1에 대한 평행하지 않은 두 평면을 결정할 수 있고, 새로운 데이터 $x \in R^n$에 대해 어떤 클래스에 속하는지를 결정하는 결정 평면은 다음과 같이 나타 낼 수 있다.

$$x^T w^{(r)} + b^{(r)} = \underset{l=1,2}{Min} | x^T w^{(l)} + b^{(l)} | \tag{16}$$

3. Fuzzy Twin Support Vector Machine

3.1 Fuzzy Membership을 포함한 데이터 집합

Fuzzy Membership을 포함한 학습 데이터들의 집합을 다음과 같다고 가정한다. [2], [3], [4]

$$(y_1, x_1, s_1), \cdot\cdot\cdot, (y_m, x_m, s_m) \tag{17}$$

x_i는 학습 데이터, y_i는 +1 또는 -1로써 데이터가 속한 클래스를 의미하며, s_i는 각 데이터마다 부여되는 Fuzzy Membership으로써 $0 \le s_i \le 1$의 범위를 갖는다.

위의 데이터 집합을 기반으로 하여 TWSVM1의 라그랑지안 함수를 재구성하면 다음과 같다.

$$L(w^{(1)}, b^{(1)}, q, \alpha, \beta) = \frac{1}{2}(Aw^{(1)} + e_1 b^{(1)})^T (Aw^{(1)} + e_1 b^{(1)}) + c_1 s_{+1,i} e_2^T q - \alpha^T(-(Bw^{(1)} + e_2 b^{(1)}) + q - e_2)) - \beta^T q \tag{18}$$

또한 TWSVM1의 최적화 함수에 $s_{+1,i}$를 추가함으로써 식 (13)의 제약식 즉, α의 범위가 다음과 같이 바뀌게 된다.

$$0 \le \alpha \le s_{+1,i} c_1 \tag{19}$$

즉, 식 (13)과 (14)를 재구성하면 다음과 같다.

$$\underset{\alpha}{Max}\ e_2^T \alpha - \frac{1}{2}\alpha^T (G(H^T H)^{-1} G)\alpha \tag{20}$$
$$0 \le \alpha \le s_{+1,i} c_1$$
$$\underset{\alpha}{Max}\ e_1^T \gamma - \frac{1}{2}\gamma^T (H(G^T G)^{-1} H)\gamma \tag{21}$$
$$0 \le \gamma \le s_{-1,i} c_2$$

3.2 Fuzzy Membership의 결정

TWSVM의 식(1), (2)에 포함된 변수 c_1, c_2는 앞서 언급한 바와 같이 식 (1)과 (2)의 최소화 과정에서 데이터와 평면 사이 거리 합의 최소화와 에러 변수의 최소화 사이의 중요도를 결정하는 변수이다.

이에 반해, 식(18)에 포함된 s_i는 두 종류의 클래스로 데이터를 분류하는데 있어, 데이터 자체의 중요도를 결정하는 변수이다. 즉, 앞서 제시한 일반적인 TWSVM은 모든 데이터가 동일한 중요도를 가지는 반면, Fuzzy Membership을 포함한 TWSVM은 각 데이터마다 중요도가 부가되어 잘못 분류된 데이터와 실제 속한 클래스에 해당하는 평면과의 단순 거리가 아닌 잘못 분류된 데이터의 중요도를 함께 포함하여 최소화하게 된다.

본 논문에서는 각 클래스의 평균값에서 가까울 수 록 s_i에 큰 수를 적용하는 방식을 선택하였으며, 그 값을 정하는 방식은 다음과 같다.

클래스 +1 데이터 값의 평균을 x_+, 클래스 -1 데이터 값의 평균을 x_-라 한다. 이 때, 클래스 +1과 클래스 -1에 대해 각 클래스의 데이터를 모두 포함하는 원의 반지름은 다음과 같다. [4]

클래스 +1 : $\quad r_+ = \underset{\{x_i : y=+1\}}{Max} | x_+ - x_i |$

클래스 -1 : $\quad r_- = \underset{\{x_i : y=-1\}}{Max} | x_- - x_i |$

위의 결과를 이용하여, 각 데이터에 해당하는 Fuzzy Membership을 다음과 같이 적용한다.

$$s_{+1,i} = 1 - | x_+ - x_i | / (r_+ + \delta) \tag{22}$$
$$s_{-1,i} = 1 - | x_- - x_i | / (r_- + \delta) \tag{23}$$

식 (22)와 (23)의 δ는 r_+ 또는 r_-가 0이 될 경우를 고려하여 추가된 상수이며, 위와 같이 결정된 $s_{+1,i}$과 $s_{-1,i}$는 0보다 크고, 1보다 같거나 작은 값을 갖게 된다.

4. 시뮬레이션을 통한 TWSVM과 Fuzzy TWSVM의 비교

시뮬레이션에 적용될 학습 데이터로써 2차원 벡터를 사용하였다. ($x = [x_1\ \ x_2]$) 클래스 +1의 데이터는 [0 0]를 중심으로 x_1, x_2를 각각 0부터 10까지 랜덤하게 발생시킨 벡터를 사용하였으며, 클래스 -1의 데이터는 [20 20]을 중심으로 x_1, x_2를 각각 0부터 10까지 랜덤하게 발생시킨 벡터를 사용하였다.

그림 1과 그림 2는 같은 데이터를 각각 TWSVM과 Fuzzy TWSVM을 사용하여 분류한 결과를 보여준다.

두 분류기의 성능 비교를 위해 클래스 +1과 클래스 -1 각각 300개의 학습 데이터를 생성하고, 동일한 데이터를 두 분류기에 적용하여 분류 성공률을 비교하였다.

이 때, TWSVM의 c_1과 c_2는 각각 100으로 지정하고, Fuzzy TWSVM의 c_1과 c_2의 경우는 잘

못 분류된 데이터의 중요도를 최대한 TWSVM과 비슷하게 하여 비교하기 위해 100을 클래스 +1과 클래스 -1 데이터의 s_i값의 평균으로 나눈 값으로 지정하였다. 즉, Fuzzy TWSVM의 $c_1 s_{+1,i}$과 $c_2 s_{-1,i}$ 평균값이 TWSVM의 c_1, c_2와 최대한 동일하게 지정하였다. 다음의 표 1은 앞서 설명한 실험의 결과를 보여준다.

FTWSVM	91.5	92.5	91.17	91.17	93	91.5	90.17	92	90.83	91.3
TWSVM	91.3	92.5	91.17	90.83	93	91.17	89.83	91.67	91	91.3

표 1 FTWSVM과 TWSVM 학습 데이터 분류 성공률 (단위 %)

실험 결과 Fuzzy TWSVM 분류기는 TWSVM 분류기보다 좋은 결과를 보여준다. 연속적인 10번의 실험에서 6차례가 Fuzzy TWSVM의 학습 데이터 분류 성공률이 높게 나왔으며, 3차례가 동일한 성공률을 보였다. 평균적인 성공률은 Fuzzy TWSVM의 경우 91.514%, TWSVM의 경우는 91.377%로써 Fuzzy TWSVM의 경우가 600개 데이터 기준으로 한 번의 수행에 대해 약 0.85개 정도의 에러 데이터의 수가 적다고 볼 수 있다.

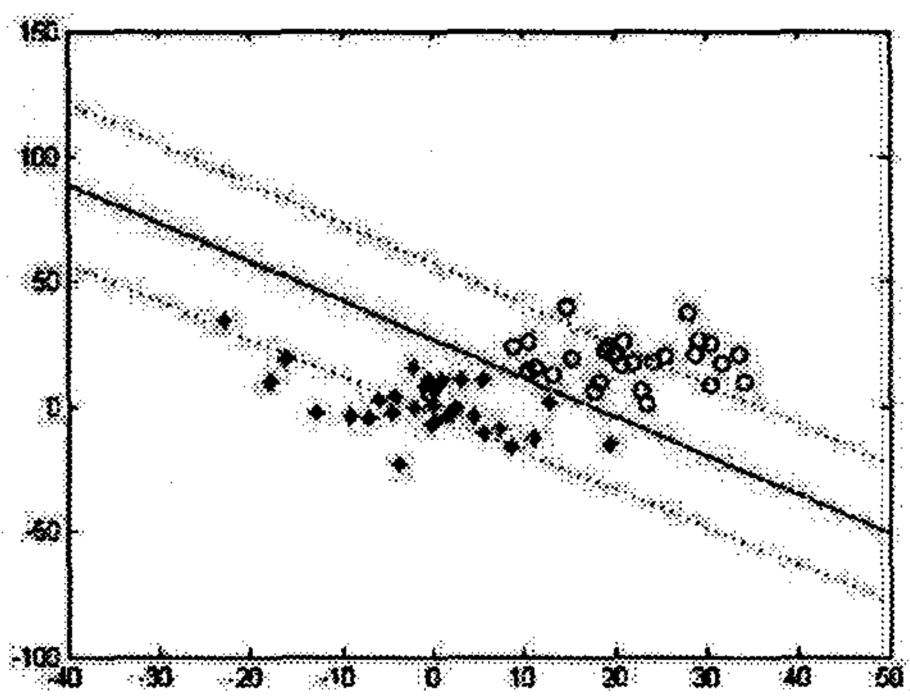

그림 1 TWSVM을 통한 데이터 분류

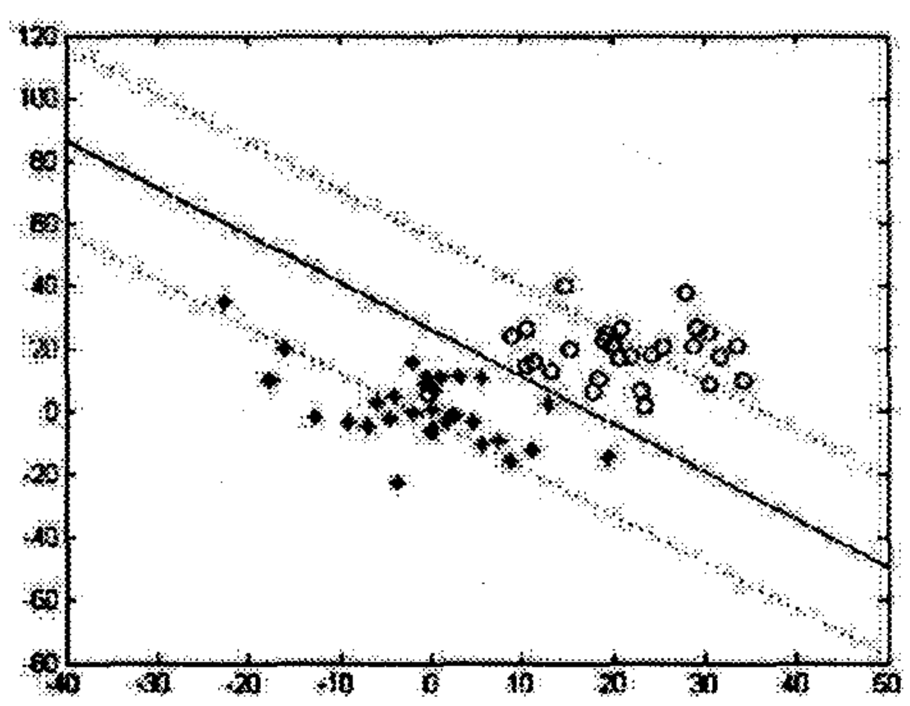

그림 2 Fuzzy TWSVM을 통한 데이터 분류

4. 결론

본 논문에서 제시한 Fuzzy Twin Support Vector Machine은 기존의 Twin Support Vector Machine과 동일하게 Support Vector Machine에 비해 약 4배 빠른 학습 속도를 유지하면서 각 학습 데이터마다 Fuzzy Membership을 적용하여 기존의 Twin Support Vector Machine에 비해 높은 학습 데이터의 분류 성공률을 보여주었다.

이와 같은 결과를 통해 영상 내에서의 패턴 분류와 같은 적용에서도 빠른 학습 속도와 이전 분류기에 비해 좋은 결과를 기대할 수 있다.

<감사의 글>
본 연구는 한국과학재단 특정기초연구(R01-2006-000-11016-0)지원으로 수행되었음.

참 고 문 헌

[1] Jayadeva and R. Khemchandani, "Twin Support Vector Machines for Pattern Classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 5, pp 905-910, May 2007

[2] Yongqiao Wang, Shouyang Wang and K. K. Lai, "A New Fuzzy Support Vector Machine to Evaluate Credit Risk," IEEE Transactions on Fuzzy Systems, Vol. 13, No. 16, pp 820-831, December 2005

[3] Yi-Hung Liu and Yen-Ting Chen, "Face Recognition Using Total Margin-Based Adaptive Fuzzy Support Vector Machines," IEEE Transactions on Neural Networks, Vol. 18, No, 1, pp 178-192, January 2007

[4] Chun-Fu Lin, and Sheng-De Wang, "Fuzzy Support Vector Machines," IEEE Transactions on Neural Networks, Vol. 13, No. 2, pp 464- 471, March 2002.

[5] Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification," Wiley, pp 215-281