

회귀용 Support Vector Machine의 성능개선을 위한 조합형 학습알고리즘

조 용 현[†] · 박 창 환^{††} · 박 용 수^{†††}

요 약

본 논문에서는 회귀용 support vector machine의 성능 개선을 위해 모멘텀과 kernel-adatron 기법이 조합된 조합형 학습알고리즘을 제안하였다. 제안된 학습알고리즘은 support vector machine의 학습기법인 기울기상승법에서 발생하는 최적해로의 수렴에 따른 발진을 억제하여 그 수렴속도를 좀 더 개선시키는 모멘텀의 장점과 비선형 특징공간에서의 동작과 구현의 용이성을 가진 kernel-adatron 알고리즘의 장점을 그대로 살린 것이다. 제안된 알고리즘의 support vector machine을 1차원과 2차원 비선형 함수 회귀에 적용하여 시뮬레이션한 결과, 학습속도에 있어서 2차 프로그래밍과 기존의 kernel-adatron 알고리즘보다 더 우수하고, 회귀성능면에도 우수한 성능이 있음을 확인하였다.

Hybrid Learning Algorithm for Improving Performance of Regression Support Vector Machine

Yong-Hyun Cho[†] · Chang-Hwan Park^{††} · Yong-Su Park^{†††}

ABSTRACT

This paper proposes a hybrid learning algorithm combined momentum and kernel-adatron for improving the performance of regression support vector machine. The momentum is utilized for high-speed convergence by restraining the oscillation in the process of converging to the optimal solution, and the kernel-adatron algorithm is also utilized for the capability of working in nonlinear feature spaces and the simple implementation. The proposed algorithm has been applied to the 1-dimension and 2-dimension nonlinear function regression problems. The simulation results show that the proposed algorithm has better the learning speed and performance of the regression, in comparison with those using quadratic programming and kernel-adatron algorithm.

키워드 : Support Vector Machines, 회귀(Regression)분석, 모멘텀(momentum), Kernel-Adatron 알고리즘, 2차 프로그래밍(Quadratic Programming) 알고리즘

1. 서 론

학습시스템에서 학습과 대규모 병렬처리의 속성을 가지는 신경망에 의한 접근은 데이터의 overfitting에 따른 모델을 생성할 수 있어 일반화측면에서 어려움이 뒤따르게 된다. 이는 학습을 위한 최적화 알고리즘과 가장 좋은 모델을 선택하는데 이용된 통계적 척도의 결과이다. 이러한 신경망이 가지는 제약들을 해결하기 위해서 support vector machine(SVM)이 제안[1-11]되었으며, 문자, 얼굴, 그리고 물체 인식 등의 실제분야에 성공적으로 적용되었다.

SVM이 overfitting을 효과적으로 막아주는 것은 Vapnik-Chervonenkis(VC) 이론으로 설명될 수 있으며[1], 그것의 학

습은 볼록함수(convex function)를 최대화함으로써 수행되는 것으로 이는 polynomial time내에 발견될 수 있는 유일해가 존재한다는 것을 의미한다. 결국 SVM은 polynomial 학습머신, radial basis 함수 망, 그리고 다층퍼셉트론(multi-layer perceptron : MLP)[1, 2]를 위한 또 다른 학습기법으로 생각될 수 있다. 여기서 가중치를 결정하는 학습은 오목하고 조건을 포함하지 않는 최소화 문제를 푸는 것과는 달리 선형 부등과 항등 조건을 가진 2차 프로그래밍(quadratic programming : QP) 문제를 해결하는 것이다[2, 3].

최근 수많은 분야에서 SVM이 성공적으로 적용되었으나 아직까지 머신학습에서 신경망 등과는 달리 표준도구로 인정받지 못하고 있다. 이는 QP 문제를 푸는데 복잡한 계산이 요구되고 시스템의 구현에도 어려움이 있기 때문이다. 이때 요구되는 메모리량은 데이터 수의 차승에 비례하며, 학습 또한 대단히 느리게 이루어진다. 따라서 SVM의 학습

† 정 회 원 : 대구카톨릭대학교 공과대학 컴퓨터정보통신공학부 교수
†† 준 회 원 : (주)텍솔 아이엔씨
††† 준 회 원 : 대구 기동대학 및 포항1대학 교수
논문접수 : 2001년 7월 5일, 심사완료 : 2001년 8월 23일

은 복잡하고 미묘하며 때로는 구현도 어렵다. 이러한 제약 점을 해결하기 위한 많은 연구가 진행되었다[3-7]. Osuna 등은 QP 문제를 여러개의 작은 QP 문제로 나누어 학습시키는 chunking을 이용한 기법[3]과 QP 문제의 행렬을 일정한 크기로 하여 학습시키는 분할기법들을 제안[4]하였다. 하지만 여기에서는 메모리 문제와 알고리즘의 수행 시 발생하는 수치적 정확성의 문제가 따르며 이로 인한 시간적인 제약도 가지고 있다. Platt은 Osuna의 기법들을 개선시킨 순차적 최소화 기법을 제안하여 학습시간을 개선하였다[5]. 한편 Campbell 등은 kernel 방법과 퍼셉트론 규칙을 병합한 반복기법의 kernel-adatron 알고리즘을 제안하였다[7]. kernel-adatron 알고리즘은 kernel에 의한 비선형 특징공간에서의 동작과 adatron의 용이한 구현성을 조합한 하이브리드 학습기법이다. 여기서는 특징공간에서의 여유를 최대화하기 위하여 기울기상승(gradient ascent)법을 이용하고 등식의 조건을 만족시키는 라그랑지안 계수(Lagrangian multiplier)을 얻기 위해서 할선법(secant method)을 이용하였다. 그러나 기울기상승법은 해의 변화를 라그랑지안 함수의 미분에 따라 변화시킴으로서 최적해로 수렴될 때 발진이 발생할 수도 있다. 따라서 우수한 구현성은 그대로 살리면서도 최적해로의 수렴에서 발생하는 발진을 막아 좀 더 빠른 속도로 SVM을 학습시킬 수 있는 효과적인 기법의 연구가 요구된다.

본 연구에서는 기울기상승법에서 해의 변화에 과거의 속성을 반영하는 모멘텀(momentum)을 이용하는 수정된 kernel-adatron 알고리즘을 제안하고 이를 SVM의 학습기법으로 이용하였다. radial basis function(RBF)의 kernel을 이용한 제한된 학습기법의 SVM을 비선형 함수회귀에 적용하여 시뮬레이션하고 그 타당성을 확인하였으며, QP 알고리즘과 기존의 kernel-adatron 알고리즘을 이용한 결과와도 비교 고찰하였다.

2. Support Vector Machine

SVM은 기존의 신경망 등에서 이용된 경험적 위험을 최소화하는 원리보다는 구조적 위험을 최소화하는 근사적 구현이다. 구조적 위험을 최소화하는 방법은 학습오차 비율의 합으로 범위가 결정되는 시험오차 항과 학습머신의 VC-차원에 의존하는 항에 기반을 두고 있다. 여기서는 회귀문제를 대상으로 소개한다[6-11].

훈련데이터 $\{(x_i, y_i), i=1, \dots, n\}$ 가 주어졌을 때, 선형회귀 문제는 주어진 입력값 x_i 에 대해 실제 목표값 y_i 들로부터 최고 ϵ 만큼의 편차 내에 있으며 가능한 작은 크기의 가중치 벡터 w 값을 갖는 다음과 같은 함수 $f(x)$ 를 찾는 것이다.

$$f(x) = w^T x_i + b \tag{1}$$

여기서 T 는 벡터의 전치이며, b 는 바이어스이다. 이 식은 가중치 변화에 의해 함수근사를 하는 다층전향 신경망에서도 활용되고 있다. 이를 위한 한가지 방법은 유클리드 놈(Euclidean norm) $\|w\|^2$ 을 최소화하는 것으로 다음과 같은 볼록 최적화 문제로 간주될 수 있다. 즉,

$$\begin{aligned} &\text{minimize } \frac{1}{2} \|w\|^2 & (2) \\ &\text{subject to } \begin{cases} y_i - w^T x_i - b \leq \epsilon \\ w^T x_i + b - y_i \leq \epsilon \end{cases} \end{aligned}$$

이다. 이 문제를 해결하기 위해 새로운 슬랙변수 ξ_i 와 ξ_i^* 를 도입하고, 이들을 포함하는 새로운 최적화 형태가 식 (2)로부터 유도될 수 있다. 즉,

$$\begin{aligned} &\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) & (3) \\ &\text{subject to } \begin{cases} y_i - w^T x_i - b \leq \epsilon + \xi_i \\ w^T x_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned}$$

이다. 여기서 C 는 양수 값으로 함수 f 의 평평함(flatness)과 편차 ϵ 의 허용한계의 균형에 따라 결정된다. 또한 식에서 슬랙변수 ξ_i 와 ξ_i^* 는 각각 출력에 대한 상한조건과 하한조건을 나타낸다. 따라서 식 (3)의 최적화 문제를 좀 더 쉽게 해결하기 위해 라그랑지안 계수 α_i 와 α_i^* 를 도입하면 목적 함수(objective function)와 제약조건으로 구성되는 이원문제의 라그랑지안 함수는

$$\begin{aligned} L = &\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ &- \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + w^T x_i + b) \\ &- \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* + y_i - w^T x_i - b) - \sum_{i=1}^n \eta_i \xi_i + \eta_i^* \xi_i^* \tag{4} \end{aligned}$$

로 나타낼 수 있으며, 여기서 η_i 와 η_i^* 는 학습률이다. 이때 제약조건은 $\alpha_i, \alpha_i^*, \eta_i$, 및 η_i^* 가 0을 포함한 양수값이어야 한다. 그러므로 식 (4)를 최소화하는 것은 L 의 요소 b, w , 그리고 $\xi_i^{(*)}$ 에 대해서 각각 미분하면 얻어질 수 있다. 즉,

$$\partial_b L = \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \tag{5}$$

$$\partial_w L = w - \sum_{i=1}^n (\alpha_i - \alpha_i^*) x_i = 0 \tag{6}$$

$$\partial_{\xi_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{7}$$

가 된다. 얻어진 식 (5)에서 식 (7)까지를 식 (4)에 대입하여 정리하면 라그랑지안 함수와 그 제약조건은

$$\begin{aligned} \text{maximize} \quad & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) x_i^T x_j \\ & - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \quad (8) \\ \text{subject to} \quad & \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned}$$

이다. 여기서 제약조건이 만족되는 방정식을 푸는 것은 식 (8)을 최대화하는 라그랑지안 계수 α_i 와 α_i^* 을 구하는 것이다. 따라서 구해진 최적의 계수 $\overline{\alpha}_i$ 와 $\overline{\alpha}_i^*$ 로부터 최적의 가중치 벡터 \overline{w} 와 바이어스 \overline{b} 는 각각 식 (6)과 support vector로부터

$$\begin{aligned} \overline{w} &= \sum_{i=1}^n (\overline{\alpha}_i - \overline{\alpha}_i^*) x_i \\ \overline{b} &= -\frac{1}{2} \overline{w}^T (x_r + x_s) \quad (9) \end{aligned}$$

와 같이 얻어진다. 여기서 x_r 과 x_s 는 각각 $\overline{\alpha}_r > 0$ 와 $\overline{\alpha}_s > 0$ 이고, $y_r = 1$ 이며 $y_s = -1$ 인 support vector들이다. 구해진 \overline{w} 와 \overline{b} 를 이용한 SMV의 회귀함수 $f(x)$ 를 정리하면

$$f(x) = \sum_{i=1}^n (\overline{\alpha}_i - \overline{\alpha}_i^*) x_i^T x + \overline{b} \quad (10)$$

이다.

이상에서 살펴 본 회귀함수는 선형의 경우에만 국한하여 설명한 것으로 모든 문제에 적용될 수는 없다. 그러므로 비선형 회귀함수도 가능하게 하기 위해서는 좀 더 일반적인 문제에도 적용될 수 있는 SVM이 고려되어야 한다. 이를 위한 방법으로 SVM에서 입력벡터 x 를 고차원의 특징공간 z 로의 사상을 이용한다. 그러나 이 방법은 고차원 특징공간에서의 내적 ($z(x) \cdot z(x_i)$)의 계산을 요구한다. 어떤 조건 하에서, 내적 계산은 아주 비효율적이지만 kernel 함수 K 를 사용함으로써 효율적인 계산이 가능하다. 여기서 kernel 함수와 특징공간과의 관계는 다음과 같다. 즉,

$$K(x, x_i) = z(x)^T \cdot z(x_i) \quad (11)$$

이며, 이때 사상을 위한 kernel 함수로는 polynomial 함수, RBF 함수, 그리고 s-자형 함수 등이 이용된다.

한편, ε -insensitive 손실함수를 사용하는 SVM의 비선형 회귀함수의 해는 다음의 식으로 구해진다. 즉,

$$\begin{aligned} \text{maximize} \quad & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\ & - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \quad (12) \\ \text{subject to} \quad & \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned}$$

이다. 이는 선형 회귀함수의 표현 식 (8)에서 입력벡터의 내적을 kernel 함수로 대체한 것에 불과하다. 여기에서도 방정식을 푸는 것은 식 (12)를 최대화하는 라그랑지안 계수 α_i 와 α_i^* 를 구하는 것으로 선형 회귀함수의 경우와 동일하다. 따라서 구해진 라그랑지안 계수 α_i 와 α_i^* 를 이용한 최적의 가중치 벡터 \overline{w} 와 바이어스 \overline{b} 는 각각

$$\begin{aligned} \overline{w} &= \sum_{i \in SVs} (\overline{\alpha}_i - \overline{\alpha}_i^*) z(x_i), \\ \overline{b} &= -\frac{1}{2} \sum_{i \in SVs} (\overline{\alpha}_i - \overline{\alpha}_i^*) [K(x_r, x_i), K(x_s, x_i)] \quad (13) \end{aligned}$$

로 계산될 수 있다. 또한 계산된 \overline{w} 와 \overline{b} 을 이용한 특징공간에서의 최적의 회귀함수는 다음과 같이 나타낼 수 있다. 즉,

$$f(x) = \sum_{i \in SVs} (\overline{\alpha}_i - \overline{\alpha}_i^*) K(x_i, x) + \overline{b} \quad (14)$$

이다.

지금까지 살펴본 SVM을 이용한 최적의 회귀함수를 구하는 문제는 최적화 패키지를 사용하여 해결할 수 있는 QP 문제로 변환하여 해결하는 것이다. 그러나 QP 과정에서는 계산의 복잡성과 구현의 어려움은 그대로 남아 있으며, 문제의 규모가 커질수록 그 제약은 더욱 더 심화된다. 결국 좀 더 빠르고 구현이 용이한 새로운 해결책이 요구된다.

3. 모멘텀과 Kernel-Adatron이 조합된 학습알고리즘

kernel-adatron 알고리즘은 퍼셉트론과 유사한 알고리즘인 adatron과 비선형 특징공간에서의 동작을 위한 kernel을 조합한 조합형 학습기법이다[7]. adatron은 이론적으로 최적해로의 수렴이 알고리즘의 반복에 지수 함수적으로 빠르게 일어난다고 알려져 있다. 결국 이 기법은 SVM의 특징공간에 adatron을 도입하는 것으로 용이한 구현성뿐만 아니라 kernel에 의한 비선형 특징공간에서의 단순한 동작 특성도 동시에 얻을 수 있어, 기존 QP 알고리즘이 가지는 계산과 구현상의 제약들을 효과적으로 해결할 수 있다.

라그랑지안 함수의 미분을 둔 기울기상승법을 사용하여 식 (12)의 라그랑지안 함수를 최대화시키는 kernel-adatron 알고리즘을 유도한다. 이는 변수 α_i 와 α_i^* 에 대한 2차 함수로 표현되는 등식의 선형제약 조건을 가지는 최적화 문제인 표준 QP 문제이다. 이때 라그랑지안 함수 L 은

$$\begin{aligned} L &= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\ & - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \quad (15) \end{aligned}$$

로 표현된다. 함수 L 의 최대화는 미분을 기초를 둔 기울기

상승법으로 가능하며, 이때 최대화를 위한 α 와 α^* 의 변화량 $\delta\alpha$ 와 $\delta\alpha^*$ 를 각각 구하는 식은

$$\begin{aligned} \delta\alpha_k(t+1) &= \eta_1 \frac{\partial L}{\partial \alpha_k(t)} \\ &= \eta_1 \left(- \sum_{i=1}^n (a_i(t) - \alpha_i^*(t)) K(x_k, x_i) - \varepsilon + y_k \right) \\ \delta\alpha_k^*(t+1) &= \eta_2 \frac{\partial L}{\partial \alpha_k^*(t)} \\ &= \eta_2 \left(- \sum_{i=1}^n (a_i(t) - \alpha_i^*(t)) K(x_k, x_i) - \varepsilon - y_k \right) \end{aligned} \quad (16)$$

이다. 여기서 t 는 반복수(epoch)이다.

하지만 식 (16)에서는 α 와 α^* 의 변화량 $\delta\alpha_k$ 와 $\delta\alpha_k^*$ 의 계산에 기울기상승법을 그대로 이용하고 있어 최적해로 수렴될 때 학습률 η 에 따라서는 발진이 일어날 수도 있다. 따라서 수렴에 따른 발진을 막아 줄 수만 있다면 좀 더 빠르게 최적해로 수렴시킬 수 있을 것이다. 이를 위해 본 연구에서는 $\delta\alpha_k$ 와 $\delta\alpha_k^*$ 를 계산할 때 이전 변화의 속성을 반영하는 모멘텀을 추가하였다. 이는 현재의 계산방향이 이전의 계산방향을 따르도록 하는 것이다. 이렇게 하면 kernel-adatron 알고리즘이 가지는 우수한 구현성은 그대로 살리면서도 최적해로의 수렴에서 일어나는 발진을 억제시켜 좀 더 빠른 속도로 SVM을 학습시킬 수 있다. 제안된 모멘텀을 이용한 $\delta\alpha_k$ 와 $\delta\alpha_k^*$ 의 계산식은 다음과 같다.

$$\begin{aligned} \delta\alpha_k(t+1) &= \eta_1 \frac{\partial L}{\partial \alpha_k(t)} \\ &= \eta_1 \left(- \sum_{i=1}^n (a_i(t) - \alpha_i^*(t)) K(x_k, x_i) - \varepsilon + y_k \right) \\ &\quad + m(\alpha_k(t) - \alpha_k(t-1)) \\ \delta\alpha_k^*(t+1) &= \eta_2 \frac{\partial L}{\partial \alpha_k^*(t)} \\ &= \eta_2 \left(- \sum_{i=1}^n (a_i(t) - \alpha_i^*(t)) K(x_k, x_i) - \varepsilon - y_k \right) \\ &\quad + m(\alpha_k^*(t) - \alpha_k^*(t-1)) \end{aligned} \quad (17)$$

여기서의 m 은 모멘텀이다.

이상에서 살펴본 제안된 알고리즘의 수행과정을 정리하면 다음과 같다.

- 단계 1: 라그랑지안 초기 계수 $\alpha_i(0), \alpha_i^*(0)$, 학습률 η , 모멘텀 m 및 최대반복수 t_{max} 를 설정한다.
- 단계 2: $i, j = 1$ 에서 n 까지 kernel 함수 $K(x_i, x_j)$ 를 계산한다.
- 단계 3: 알고리즘의 반복수 $t = 1$ 에서 t_{max} 까지 다음 단계 4에서 6까지를 수행한다.
- 단계 4: $\delta\alpha_k(t)$ 와 $\delta\alpha_k^*(t)$ 를 계산하고

만약 $(\alpha_k^*(t) + \delta\alpha_k^*(t)) \leq 0$ 이면,

$$\alpha_k^*(t) = 0 \text{으로,}$$

만약 $0 < (\alpha_k^*(t) + \delta\alpha_k^*(t)) < C$ 이면,

$$\alpha_k^*(t) = \alpha_k^*(t) + \delta\alpha_k^*(t) \text{로}$$

만약 $(\alpha_k^*(t) + \delta\alpha_k^*(t)) \geq C$ 이면,

$$\alpha_k^*(t) = C \text{로 한다.}$$

단계 5: \bar{w} 와 \bar{b} 를 계산한다.

단계 6: 만약 반복수 t_{max} 를 넘거나 여유 x 를 계산하여 근사적으로 1이면 학습을 종료하고, 그렇지 않으면 단계 3으로 간다.

$$\begin{aligned} x = \frac{1}{2} \left[\min_{(i \setminus y_i = 1, \alpha_i \leq c)} \sum_{i=1}^n (a_i - \alpha_i^*) K(x_i, x_j) \right. \\ \left. - \max_{(i \setminus y_i = -1, \alpha_i \leq c)} \sum_{i=1}^n (a_i - \alpha_i^*) K(x_i, x_j) - y_i \right] \end{aligned}$$

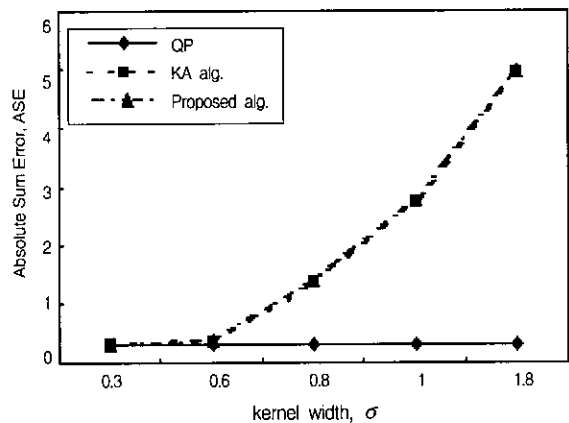
4. 시뮬레이션 결과 및 분석

제안된 알고리즘의 회귀용 SVM의 타당성과 성능을 평가하기 위해서 1차원 및 2차원 비선형 회귀함수를 대상으로 RBF kernel 함수 $K(x, x_i) = \exp(-\|x - x_i\|^2 / 2\sigma^2)$ 을 이용하여 시뮬레이션 하였다. 시뮬레이션 결과는 표준 QP 알고리즘 및 기존의 kernel-adatron 알고리즘의 결과와 비교 분석하였다.

4.1 1차원 비선형 회귀분석

1차원 비선형 함수 $f(x) = \sin(x)$ 을 대상으로 x 의 값은 -10에서 +10까지의 구간에서 0을 포함한 0.4의 동간격으로 51개 데이터를 추출하여 실험하였다.

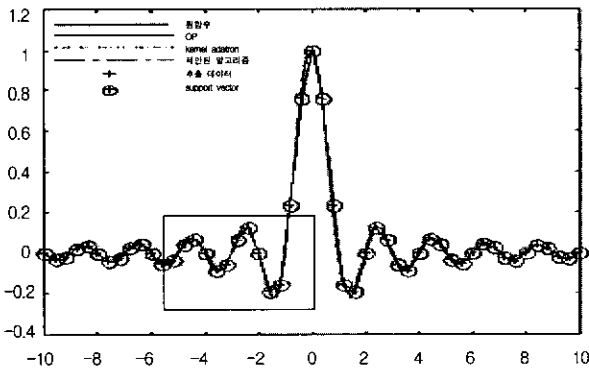
(그림 1)은 $\varepsilon=0, \eta=0.5$, 제안된 알고리즘의 모멘텀 $m=0.01$ 로 하여 radial basis 함수의 폭 σ 의 변화에 따른



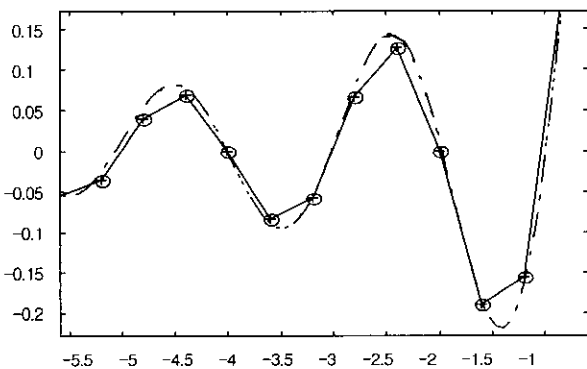
(그림 1) Radial basis 함수의 폭 σ 의 변화에 따른 절대합오차 ASE

실제 데이터 값과 근사화된 값과의 절대합오차(Absolute Sum Error : ASE) = $\sum_{i=1}^m |y_i - f(x)|$ 를 나타낸 것이다. 그림으로부터 QP 알고리즘은 σ 값에 무관하게 일정한 ASE 값을 가지나 kernel adatron 알고리즘과 제안된 알고리즘은 상대적으로 QP 알고리즘보다 kernel 폭 σ 에 더욱 의존됨을 알 수 있다. 그림에서 제안된 알고리즘과 kernel-adatron 알고리즘의 경우, σ 의 값은 0.3에서 0.6 사이가 가장 적당함을 알 수가 있다.

(그림 2)는 $\epsilon=0, \eta=0.5, \sigma=0.6, m=0.01$ 로 하여 대상 함수에 대한 QP 알고리즘, kernel-adatron 알고리즘, 그리고 제안된 알고리즘의 회귀결과를 보여 주는 그래프이다. 그림에서 kernel-adatron 알고리즘과 제안된 알고리즘은 거의 유사하게 근사화됨을 알 수 있다. 하지만 QP 알고리즘의 경우는 상대적으로 근사화 정도가 떨어짐을 알 수가 있다. 그림의 사각부분을 확대한 (그림 3)을 보면 더욱 정확하게 알 수가 있는데, 특히 함수의 변화가 심한 양쪽 끝부분을 살펴보면 kernel-adatron 알고리즘과 제안된 알고리즘은 비교적 원함수와 유사하게 근사화되는 것에 비해, QP 알고리즘의 경우는 추출된 데이터에 선형적으로 근사화되는 것을 볼 수 있다. 즉, 추출데이터의 수가 많은 구간에서는 QP 알고리즘보다 kernel-adatron 알고리즘과 제안된 알고리즘이 더 우수한 성능이 있음을 알 수 있다.



(그림 2) $f(x) = \text{sinc}(x)$ 에 대한 QP, kernel-adatron, 제안된 알고리즘 각각의 근사화 결과



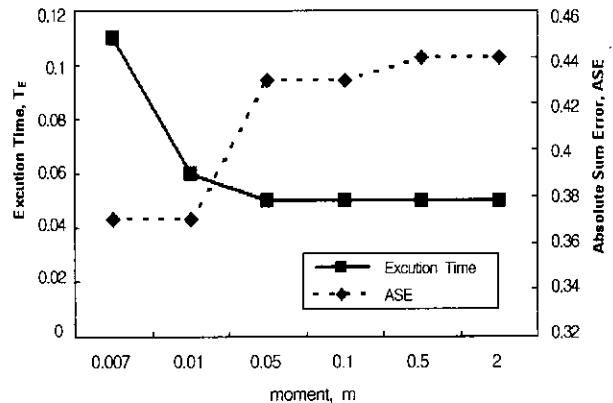
(그림 3) (그림 2)의 사각영역 확대도

한편, <표 1>은 (그림 2)의 실험조건하에서 3가지 알고리즘에 대한 학습시간 T_E 와 절대합오차 ASE를 측정 한 결과이다. 표에서 보면 제안된 알고리즘은 kernel-adatron 알고리즘에 비해 약 2.9배 정도와 QP 알고리즘에 비해 약 82배 정도 학습속도가 개선됨을 알 수 있다. 또한 kernel-adatron 알고리즘과 제안된 알고리즘만을 비교하면 ASE로 표현되는 회귀성능은 유사하나 학습속도에서는 제안된 알고리즘이 더 우수한 성능이 있다.

<표 1> $f(x) = \text{sinc}(x)$ 에 대한 학습시간과 절대합오차

성능 \ 알고리즘	QP	kernel-adatron	Proposed algorithm
학습시간 T_E	4.9	0.17	0.06
절대합오차 ASE	0.31	0.36	0.37

(그림 4)는 모멘텀 m 의 변화에 따른 제안된 알고리즘의 학습시간 T_E 와 절대합오차 ASE를 나타낸 것이다. 그림에서 보면 m 이 0.01일 때 학습시간과 절대합오차에서 가장 우수한 성능을 보인다. 하지만 다른 모멘텀 값에서도 학습시간과 오차는 제안된 알고리즘이 kernel-adatron 알고리즘보다 우수한 성능이 있음을 알 수 있다. 이는 동일 문제에 대해서 kernel-adatron 알고리즘의 경우 <표 1>에서 보면 학습시간이 0.17인데 비해 그림의 제안된 알고리즘에서는 모멘텀의 값에 따라 변하지만 m 이 0.001에서 2까지의 범위에서는 $m=0.001$ 일 때가 가장 오랜 학습시간인 0.11로 약 0.06 정도의 빠른 학습시간 차이를 알 수 있다.



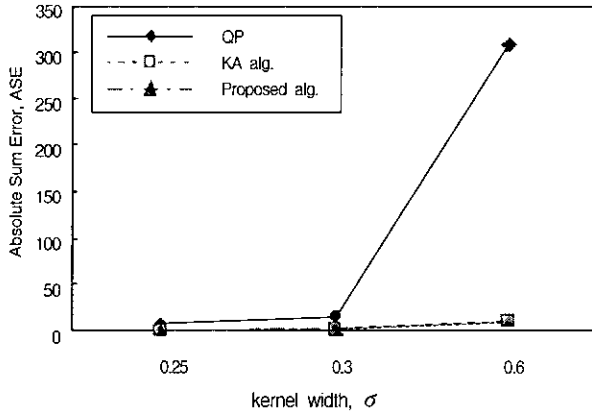
(그림 4) 모멘텀의 변화에 따른 제안된 알고리즘의 학습시간과 ASE

4.2 2차원 비선형 회귀분석

좀 더 복잡한 문제에 제안된 알고리즘을 이용한 회귀용 SVM의 성능을 평가하기 위해서 2차원 비선형 함수 $f(x, y) = \text{sinc}\sqrt{(x^2 + y^2)}$ 를 대상으로 x 와 y 각각의 구간 -9 에서 9 사이의 구간에서 0을 포함한 0.2의 등간격으로 91개 데이터를 추출하여 사용하였다.

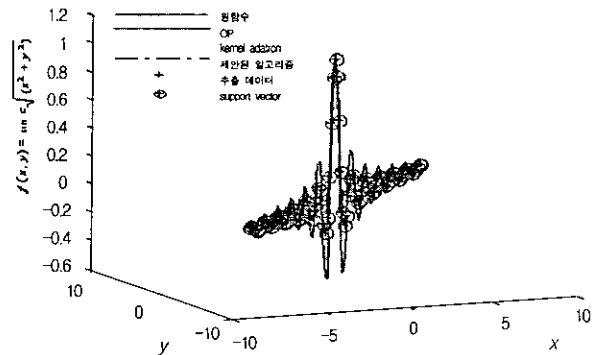
(그림 5)는 $\epsilon=0, \eta=0.08$ 제안된 알고리즘의 모멘텀 $m = 0.01$ 로 하여 radial-basis 함수의 폭 σ 의 변화에 따른

ASE를 나타낸 것이다. 그림으로부터 QP 알고리즘이 오히려 σ 값에 매우 민감함을 알 수 있고 상대적으로 kernel-adatron 알고리즘과 제안된 알고리즘은 매우 작은 ASE 값을 가지며 σ 의 값에 거의 영향을 받지 않고 인정한다는 것을 알 수 있다. 앞의 1차원 실험과 비교하여 보면 차원이 높아질수록 QP 알고리즘은 σ 에 많이 의존을 하고, 반면 kernel-adatron과 제안된 알고리즘은 σ 에 덜 의존함을 알 수 있다.

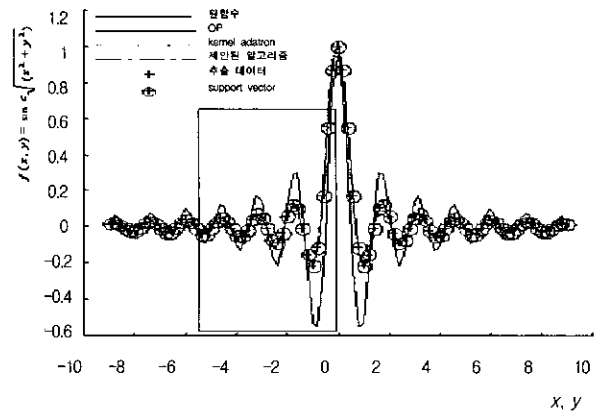


(그림 5) Radial basis 함수의 폭 σ 변화에 따른 ASE

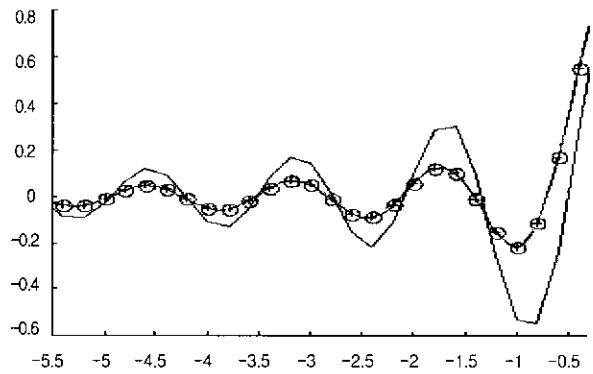
(그림 6)은 $\sigma=0.25$ 로 대상 함수에 대한 QP 알고리즘, kernel-adatron, 그리고 제안된 알고리즘의 근사화 결과를 3차원으로 보여주는 그래프이다. 확실한 구분을 위해서 (그림 6)을 방위각 90도와 고도 0으로 회전하여 (그림 7)로 나타내었다. 그림에서도 kernel-adatron과 제안된 알고리즘은 거의 유사한 근사화 결과를 보이나 QP 알고리즘은 상대적으로 근사화 정도가 떨어지는 결과를 보인다. 여기서도 1차원 함수의 실험과 동일한 결과를 나타냄을 확인할 수 있다. (그림 7)의 사각부분을 확대한 (그림 8)을 보면 더 정확하게 알 수가 있는데, 특히 함수의 x 와 y 의 구간이 -2에서 2사이에서의 근사화 결과를 보면 kernel-adatron 알고리즘과 제안된 알고리즘은 원함수와 거의 유사한 근사화 성능을 보이는 반면, QP 알고리즘의 경우는 추출된 데이터에 거의 선형적으로 근사화되어 확연한 근사화 성능차를 볼 수 있다.



(그림 6) $f(x, y) = \sin c \sqrt{x^2 + y^2}$ 에 대한 QP, kernel-adatron, 제안된 알고리즘 각각의 근사화 결과



(그림 7) 방위각 90도 및 고도 0으로 본 (그림 6)의 결과



(그림 8) (그림 7)의 사각부분 확대도

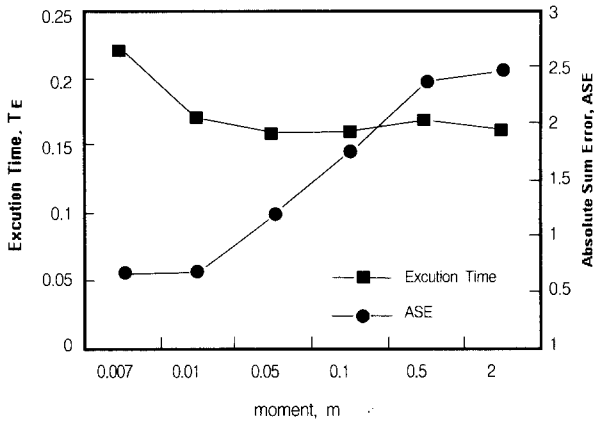
한편, <표 2>도 (그림 7)의 실험조건하에서 SVM의 학습 시간과 절대합오차를 측정된 결과이다. 제안된 알고리즘은 kernel-adatron에 비해 약 1.3배, QP에 비해 약 191배정도 학습속도가 개선되었음을 알 수 있다. ASE로 표현되는 제안된 알고리즘의 회귀성능은 kernel-adatron과는 유사하나 QP와 비교하면 훨씬 개선되었음을 알 수 있다.

<표 2> $f(x, y) = \sin c \sqrt{x^2 + y^2}$ 에 대한 학습시간과 절대합오차

성능 \ 알고리즘	QP	kernel-adatron	Proposed algorithm
학습시간 TE	32.5	0.22	0.17
절대합오차 ASE	7.31	0.639	0.679

(그림 9)는 모멘텀 m 의 변화에 따른 제안된 알고리즘의 학습시간 TE와 절대합오차 ASE를 나타낸 것이다. 그림에서 보면 m 이 0.01일 때 TE와 ASE에서 가장 우수한 성능을 보임을 알 수 있다. 또한 다른 모멘텀 값에서도 학습 시간과 오차측면에서 제안된 알고리즘은 kernel-adatron 알고리즘보다 우수한 성능이 있음을 알 수 있다. 이는 <표 2>의 결과와 비교할 때, 제안된 알고리즘의 경우 모멘텀의 값이 0.007일 때만 kernel-adatron 알고리즘과 같은 학습속도

를 보이고, 그 외의 모멘텀 값에서는 평균적으로 약 0.17 정도로 약 0.05정도의 빠른 학습시간 차이를 보인다.



(그림 9) 모멘텀의 변화에 따른 제안된 알고리즘의 학습시간과 ASE

이상의 실험 결과들로부터 제안된 방법은 QP 알고리즘이 가지는 학습속도와 구현의 어려움 및 기존의 kernel-adatron 알고리즘에서 일어나는 발진현상을 동시에 해결할 수 있음을 확인할 수 있다. 특히, 1 차원과 2 차원 비선형 회귀분석을 각각 비교할 때 제안된 알고리즘은 차원이 높고 규모가 큰 문제일수록 더욱 더 우수한 성능개선효과가 있음을 알 수 있다.

5. 결 론

본 연구에서는 회귀용 SVM의 효율적인 학습을 위해 해의 변화에 과거의 속성을 반영하는 모멘텀을 추가한 조합형 알고리즘을 제안하였다. 이렇게 하면 기울기상승법에서 발생하는 최적해로의 수렴에 따른 발진을 억제하여 그 수렴속도를 좀 더 개선시키는 모멘텀의 장점과 kernel-adatron 알고리즘의 구현 용이성을 그대로 살릴 수 있다.

제안된 학습알고리즘의 SVM을 1차원과 2차원 비선형 회귀함수에 각각 적용하여 시뮬레이션 한 결과, QP 알고리즘과 kernel-adatron 알고리즘보다 학습시간에 있어서 우수한 결과를 보였고 회귀성능에서는 kernel-adatron과 유사한 성능을 보였다. 특히 문제의 규모가 커질수록 학습속도의 성능 개선정도는 증가됨을 확인하였다.

향후 제안된 알고리즘을 좀더 큰 규모의 문제에 적용하여 성능을 확인하여야 하며, 좀 더 다양한 분야에의 응용에 대한 연구도 뒤따라야 할 것이다.

참 고 문 헌

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.
 [2] M. O. Stitson, J. A. E. Weston, A. Gammernan, V. Vovk,

and V. Vapnik, "Theory of Support Vector Machines," *Technical report CSD-TR-96-17*, Royal Holloway, Univ. of London, May, 1998.

[3] E. E. Osuna, R. Freund, and F. Girosi, "Simple Learning Algorithms for Training Support Vector Machines," <http://lara.enm.bris.ac.uk/cig/gzipped/KA-ieee.ps.gz>.
 [4] E. E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines : An Application to Face Detection," *Proc. Computer Vision and Pattern Recognition'97*, Puerto-Rico, June, 1997.
 [5] J. C. Platt, *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, In *Advances in Kernels Methods : Support Vector Learning*, MIT Press, Cambridge, 1998.
 [6] A. J. Smola, and B. Scholkopf, "A Tutorial on Support Vector Regression," *NeuroCOLT2 Technical Report*, Neuro COLT, Oct. 1998.
 [7] C. Campbell and N. Cristianini, "Simple Learning Algorithms for Training Support Vector Machines," *Dept. of Engineering Mathematics Technical Report*, Univ. of Bristol, 1998.
 [8] S. Gunn, "Support Vector Machines for Classification and Regression," *ISIS Technical Report*, Univ. of Southampton, May, 1998.
 [9] A. Smola, "Regression Estimation with Support Vector Learning Machines," *Technische Universitat. Munchen Technical Report*, Version 1.01., Dec. 1996
 [10] V. Vapnik, *The Nature of Statistical Learn Theory*, Springer Verlag, 1995.
 [11] V. Vapnik, S. Golowich, and A. Smola, *Support vector method for function approximation, regression estimation, and signal processing*, *Advances in Neural Information Processing Systems 9*, pp.281-297, Cambridge, MIT Press, MA 1997.



조 용 현

e-mail : yhcho@cuth.cataegu.ac.kr

1979년 경북대학교 공과대학 전자공학과 (공학사)

1981년 경북대학교 대학원 전자공학과 (공학석사)

1993년 경북대학교 대학원 전자공학과 (공학박사)

1983년~1984년 삼성전자(주) 사원

1984년~1987년 한국전자통신연구소 연구원

1987년~1997년 영남이공대학 전자과 부교수

1997년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 부교수
 관심분야 : 신경망, 병렬분산처리, 신호처리, 영상인식, 교환기 등



박 창 환

e-mail : pch0702@recsol.co.kr

1999년 경일대학교 전자공학과 졸업
(학사)

2001년 대구가톨릭대학교 대학원 전산통
계학과 졸업(석사)

2001년~현재 (주)렉솔아이엔씨 근무

관심분야 : 신경망 및 퍼지이론, 이미지 프로세싱



박 용 수

e-mail : timpark@chollian.net

1979년 경북대학교 전자공학과 졸업
(학사)

1981년 경북대학교 대학원 전자공학과
졸업(석사)

1985년~1992년 LG반도체(주) 근무

1993년~1994년 반도체 엔지니어링(주) 근무

1999년~2000년 포항직업전문학교 교사

1999년~2001년 대구가톨릭 대학교 박사과정 수료

2001년~현재 대구기능대학 및 포항1대학 시간강사

관심분야 : Neural Network, Fuzzy system, 진화 연산, 이미지
프로세싱