함수근사를 위한 서포트 벡터 기계의 커널 애더트론 알고리즘

석 경 하† · 황 창 하††

요 약

함수근사는 과학과 공학분야에서 광범위하게 응용된다. 서포트 벡터 기계(support vector machine, SVM)는 원래 분류를 위해 제안되어져 문자인식, 얼굴인식 등의 응용분야에서 좋은 결과를 보여주고 있다. 최근 SVM 이론이 함수근사로 확장되어 많이 활용되려 하고 있다. 그러나 함수근사를 위한 SVM 알고리즘은 QP(quadratic programming) 문제와 관련되어 있어 계산에 시간이 걸리며 QP를 위한 패키지가 있어야 한다. 본 논문에서는 함수근사를 위해 커널-애더트론 알고리즘을 이용한 SVM을 제안하고 QP를 이용한 SVM과 성능을 비교하고자 한다.

Kernel Adatron Algorithm of Support Vector Machine for Function Approximation

Kyungha Seok† · Changha Hwang††

ABSTRACT

Function approximation from a set of input-output pairs has numerous applications in scientific and engineering areas. Support vector machine (SVM) is a new and very promising classification, regression and function approximation technique developed by Vapnik and his group at AT&T Bell Laboratories. However, it has failed to establish itself as common machine learning tool. This is partly due to the fact that this is not easy to implement, and its standard implementation requires the use of optimization package for quadratic programming (QP). In this paper we present simple iterative Kernel Adatron (KA) algorithm for function approximation and compare it with standard SVM algorithm using QP.

1. 서 론

함수근사는 과학과 공학분야에서 광범위하게 응용된다. SVM은 원래 분류(classification)를 위해 Vapnik과 공동연구자들에 의해 개발되어 문자인식, 얼굴인식 등의 응용분야에서 좋은 결과를 보여주고 있다[1, 6, 9, 10]. 최근 분류를 위한 SVM 이론이 함수근사로 확장되어 많이 활용되려 하고 있다[4, 5, 7-10]. SVM은 투사지향(projection pursuit) 알고리즘 및 신경망과 함께 독립변수가 두 개 이상일 때 비선형 함수근사를 위해 사용되는 방법이다. SVM은 VC-이론을 이용하여 예측(prediction)을 잘하고 벌칙항(penalty term)을 사용하여 과대적합(overfitting)을 피하려는데 초점을 두고 있다. 여기서 VC는 이론 창시자들의 이름인 Vapnik과 Chervonenkis 첫 글자를 따서 표현된 것이다. 한편 SVM은 고도의 비선형 구조를 파악할 수 있다. 즉, 학습되는 함수는 모든 학습점 사이를 보간하여 그 구조를 파악한다. 이때 실제상황에서는 많은 경우에 학습자료에 이상치(outlier)가 포함되게 된다. 그러므로 학

※ 정보통신부에서 지원하는 대학기초연구지원사업으로 수행하였음
† 정 회 원 인제대학교 데이터정보학과 교수
†† 정 회 원 대구가톨릭대학교 정보통계학과 교수
논문접수 : 1999년 9월 11일, 심사완료 : 2000년 5월 29일

습되는 함수는 모든 자료 사이를 보간하므로 이상치가 있는 자료의 위치까지도 보간하여, 원하지 않는 구조를 파악하게 된다. 그런데 SVM은 넓칙함을 이용하여 과대적합을 피하기 때문에 이상치에 민감하지 않고 그 영향을 최소화시키는 로버스트(robust)한 방법일 것으로 생각되어 이상치가 있는 자료에 SVM을 적용하여 이상치에 어느정도 로버스트한지를 분석, 연구하고자 한다. 한편, SVM은 볼록함수(convex function)를 최소화하여 학습이 진행되기 때문에 신경망과는 달리 유일한 해를 구할 수 있는 장점이 있다.

그러나 분류와 관련된 문자인식, 얼굴인식 등의 응용분야에서 많은 성공사례를 보여주었음에도 불구하고 SVM은 QP(quadratic programming) 문제와 관련되어 있어 계산에 시간이 오래 걸리며 QP를 위한 패키지가 있어야 하기 때문에 기계학습을 위한 표준도구로 이용되는데는 어려움이 있다 물론 분류를 위한 SVM에도 같은 문제가 있다[2, 3]. 한편 이런 단점에도 불구하고 많은 응용분야에서 좋은 실험결과를 보여주어 점점 인기가 높아가고 있다 본 논문에서는 함수근사를 위해 계산 속도가 빠른 커널 애더트론(kernel adatron, KA) 알고리즘을 이용한 SVM을 제안하고 그 성능을 QP를 이용한 SVM과 비교, 분석하고자 한다.

2. 서포트 벡터 기계를 이용한 함수근사와 문제점

SVM은 원래 분류를 위해 Vapnik과 공동연구자들에 의해 개발되었으며 많은 응용분야에서 좋은 결과를 보여주고 있어 그 인기가 점점 높아가고 있다 SVM에는 두종류가 있다. 첫째는 분류를 위한 SVM이고 둘째는 함수근사를 위한 SVM이다. 최근 분류를 위한 SVM 이론이 함수근사로 확장되어 많이 활용되려 하고 있다 그러나 분류를 위한 SVM은 함수근사를 위한 SVM의 특별한 경우로 간주된다[4, 8]. 먼저 선형 함수근사를 위한 SVM을 간단하게 알아본다. 자세한 내용은 참고문헌 [5, 7-10]에 설명되어 있다

훈련자료 $\{(\boldsymbol{x}_i, y_i),\ i=1, \cdots, n\} \subset X \times R$가 주어졌다고 가정한다. 여기서 X는 입력벡터공간 R^d를 나타낸다. 우리들의 목표는 모든 훈련자료에 대해서 실제 목표값 y_i들로부터 최고 ε만큼의 편차 내에 있으며 가능한 작은 크기의 $\boldsymbol{w}$값을 갖는 함수 $f(\boldsymbol{x})$를 찾는 것이다. 다음과 같은 함수형태를 생각한다.

$$f(\boldsymbol{x}) = \boldsymbol{w}^t \boldsymbol{x} + b \text{ with } \boldsymbol{w} \in X,\ b \in R$$

여기서 윗첨자 t는 벡터의 전치를 나타낸다. 이 이론은 가중치 감소(weight decay)를 사용하여 함수근사를 하는 다층전방향 신경망에서도 활용되고 있다. 이를 위한 한가지 방법은 유클리드 놈 $\| \boldsymbol{w} \|^2$을 최소화하는 것이다. 공식적으로 우리는 이 문제를 다음과 같은 볼록 최적화(convex optimization) 문제로 간주할 수 있다[5, 7-10]

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} \| \boldsymbol{w} \|^2, \\ &\text{subject to} \quad y_i - \boldsymbol{w}^t \boldsymbol{x}_i - b \le \varepsilon \\ &\text{and} \quad \boldsymbol{w}^t \boldsymbol{x}_i + b - y_i \le \varepsilon \end{aligned}$$

여기서 기본가정은 볼록 최적화 문제가 해결 가능하다는 것이다. 그러나 가끔 이 가정이 성립되지 않는다. 따라서 이를 해결하기 위해서 새로운 변수 ξ, ξ_i^*를 도입한다. 그러면 Vapnik[9, 10]이 제안한 다음과 같은 최적화 문제로 귀착된다.

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*), \\ &\text{subject to} \quad \begin{cases} y_i - \boldsymbol{w}^t \boldsymbol{x}_i - b \le \varepsilon + \xi_i \\ \boldsymbol{w}^t \boldsymbol{x}_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i,\ \xi_i^* \ge 0 \end{cases} \end{aligned}$$

상수 $C > 0$는 함수 f의 평평함과 편차 ε의 허용한계의 균형을 고려하여 결정된다. 여기서 ξ_i, ξ_i^*는 출력에 대한 상계제약(upper constraint)과 하계제약(lower constraint)을 나타낸다. 위의 최적화 문제는 다음과 같은 ε-무감각(ε-insensitive) 손실함수 $|\xi|_\varepsilon$를 다루는 것에 대응된다.

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

핵심 아이디어는 다음과 같은 라그랑즈(Lagrange) 함수를 만드는 것이다.

$$\begin{aligned} L = \tfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ - \sum_{i=1}^{n} \alpha_i (\varepsilon + \xi_i - y_i + \boldsymbol{w}^t \boldsymbol{x}_i + b) \\ - \sum_{i=1}^{n} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \boldsymbol{w}^t \boldsymbol{x}_i - b) \\ - \sum_{i=1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \end{aligned}$$

여기서 제약조건 $\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0$이 성립해야 한다 따라서 최적화 문제는 다음과 같이 된다.

$$\text{maximize} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\boldsymbol{x}_i^t\boldsymbol{x}_j - \varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*),$$

$$\text{subject to} \quad \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

이런 제약조건을 가진 위의 방정식을 푸는 것은 라그랑즈 배수(Lagrange multiplier) α_i, α_i^*를 구하는 것을 의미한다. 한편, 최적의 함수근사는 다음과 같이 구해진다.

$$\overline{\boldsymbol{w}} = \sum_{i=1}^{n}(\overline{\alpha}_i-\overline{\alpha}_i^*)\boldsymbol{x}_i, \quad \overline{b} = -\frac{1}{2}\overline{\boldsymbol{w}}^t[\boldsymbol{x}_r+\boldsymbol{x}_s]$$

여기서 $\boldsymbol{x}_r$과 $\boldsymbol{x}_s$는 스포트 벡터이다 그러므로 최적의 함수근사는

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}(\overline{\alpha}_i-\overline{\alpha}_i^*)\boldsymbol{x}_i^t\boldsymbol{x}+\overline{b}$$

이다. Karush-Kuhn-Tucker(KKT) 조건은 다음과 같다

$$\overline{\alpha}_i\overline{\alpha}_i^* = 0, \quad i=1, \cdots, n.$$

따라서 스포트 벡터는 라그랑즈 배수 중 하나가 0 보다 큰 입력벡터이다. .

이제 비선형 함수근사를 위한 SVM을 알아보자. 자세한 내용은 참고문헌 [5,7-10]에 설명되어 있다. 위의 방법을 비선형 함수근사의 경우로 확장하고자한다. 먼저 SVM의 학습에서 자료는 내적(dot product) $\boldsymbol{x}_i^t\boldsymbol{x}_j$을 통해서 함수에 나타난다. 이제 함수 $\Phi: R^d \rightarrow E$를 사용하여 자료를 유클리드 공간 E로 대응시킨다고 가정한다. 그러면 학습 알고리즘은 E에서 내적 $\Phi(\boldsymbol{x}_i)^t\Phi(\boldsymbol{x}_j)$을 통해 자료에 의존한다. 만일 $K(\boldsymbol{x}_i, \boldsymbol{x}_j)=\Phi(\boldsymbol{x}_i)^t\Phi(\boldsymbol{x}_j)$를 만족하는 커널 함수 K가 존재한다면 우리들은 학습 알고리즘에 단지 K만을 사용하고 Φ가 구체적으로 무엇인지를 알 필요가 없다. 일반적인 커널 함수는 다음과 같다.

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^t\boldsymbol{y}+1)^p, \quad K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$$

여기서 p와 σ^2은 커널 모수이다 커널 방법은 차원문제(curse of dimensionality)를 해결하기 위해서 사용된다. ε-무감각 손실함수를 사용하는 SVM의 비선형 함수근사의 해는 다음과 같이 구해진다.

$$\text{maximize} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \qquad (1)$$

$$\text{subject to} \quad \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]$$

이런 제약조건하에서 위의 방정식을 푸는 것은 라그랑즈 배수 α_i, α_i^*를 구하는 것이며 최적의 함수근사는 다음과 같다.

$$\overline{\boldsymbol{w}} = \sum_{i=1}^{n}(\overline{\alpha}_i-\overline{\alpha}_i^*)\Phi(\boldsymbol{x}_i),$$

$$\overline{b} = -\frac{1}{2}\sum_{i=1}^{n}(\overline{\alpha}_i-\overline{\alpha}_i^*)[K(\boldsymbol{x}_r, \boldsymbol{x}_i), K(\boldsymbol{x}_s, \boldsymbol{x}_i)]$$

따라서 최적의 함수근사는 다음과 같이 표현된다.

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}(\overline{\alpha}_i-\overline{\alpha}_i^*)K(\boldsymbol{x}_i, \boldsymbol{x})+\overline{b}$$

선형인 경우와의 차이점은 $\boldsymbol{w}$가 더 이상 정확한 형태로 주어지지 않는다는 것이다. 그러나 $\boldsymbol{w}$는 내적을 통해서 유일하게 결정된다. 한편 비선형의 경우에 최적화 문제는 입력공간이 아닌 특징공간(feature space)에서 가장 평평한 함수를 찾는 것에 대응된다. 함수근사 문제에서 세가지 모수 ε, C 및 커널 모수가 결정되어야 한다.

SVM을 사용하여 최적의 함수근사를 구하는 문제는 최적화 패키지를 사용하여 해결할 수 있는 QP 문제로 귀결된다 따라서 SVM의 학습은 어렵기로 소문난 QP 문제를 해결해야 하는 난관에 봉착하게 된다. 더욱이 표준 QP 프로그래밍 루틴은 상당한 메모리 사용을 요구한다. 그리고 자료가 많을 때 SVM은 QP 문제를 일련의 다소 단순한 QP 문제로 나누는 부가적인 기법을 필요로 한다. 따라서 실제 응용을 위해서는 충분히 빠른 학습 알고리즘이 필요하다. 특히 관측치가 100개 이상인 경우의 함수근사 문제에서 QP는 많은 계산시간을 필요로 한다.

3. 커널 애더트론(KA) 알고리즘

우리는 라그랑즈 함수의 미분에 기초를 둔 기울기 상승법(gradient ascent method)을 사용하여 식 (1)의 라그랑즈 함수를 최대화시키는 KA 알고리즘을 유도한다. 라그랑즈 함수의 미분은 다음과 같이 구해진다.

$$L(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(\boldsymbol{x}_i,\boldsymbol{x}_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*)$$

$$\delta\alpha_k = \eta_1\frac{\partial L}{\partial\alpha_k} = \eta_1\Big(-\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(\boldsymbol{x}_k,\boldsymbol{x}_i)-\varepsilon+y_k\Big)$$

$$\delta\alpha_k^* = \eta_2\frac{\partial L}{\partial\alpha_k^*} = \eta_2\Big(-\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(\boldsymbol{x}_k,\boldsymbol{x}_i)-\varepsilon-y_k\Big)$$

더욱이 음수가 되는 α_i, α_i^*를 0으로 설정하여 제약조건 $\alpha_i\geq 0$, $\alpha_i^*\geq 0$이 성립하도록 만든다. 특별한 관측치 k에 대해 최신화(updating) $\alpha_k\rightarrow\alpha_k+\delta\alpha_k$와 $\alpha_k^*\rightarrow\alpha_k^*+\delta\alpha_k^*$에 따른 라그랑즈 함수의 변화량을 고려하여 최적의 η_1, η_2를 구하는 문제를 생각해보자.

$$\begin{aligned}\delta L &= L(\alpha_k^{(*)}+\delta\alpha_k^{(*)})-L(\alpha_k^{(*)})\\ &= \delta\alpha_k\Big(-\sum_{j=1}^{n}(\alpha_j-\alpha_j^*)K(\boldsymbol{x}_k,\boldsymbol{x}_j)-\varepsilon+y_k\Big)\\ &\quad+\delta\alpha_k^*\Big(\sum_{j=1}^{n}(\alpha_j-\alpha_j^*)K(\boldsymbol{x}_k,\boldsymbol{x}_j)-\varepsilon-y_k\Big)\\ &\quad-\frac{1}{2}\left[\delta(\alpha_k-\alpha_k^*)\right]^2K(\boldsymbol{x}_k,\boldsymbol{x}_k)\end{aligned}$$

한편 문제를 간단히 하기 위하여 $\eta_1=\eta_2$로 두고 커널 함수로 $K(\boldsymbol{x},\boldsymbol{y})=e^{-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}}$를 사용하면 라그랑즈 함수의 변화량은 다음과 같이 간단하게 표현된다.

$$\begin{aligned}\delta L &= \eta\left[2c_k^2+2y_k^2-4c_ky_k+2\varepsilon^2\right]\\ &\quad-\frac{1}{2}\eta^2\left[4c_k^2+4y_k^2-8c_ky_k\right]\\ &= 2(c_k-y_k)^2(\eta-\eta^2)+2\eta\varepsilon^2\end{aligned}\qquad(2)$$

여기서 c_k는 다음과 같이

$$c_k=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(\boldsymbol{x}_k,\boldsymbol{x}_i)$$

로 주어지며 y_k의 추정치가 된다. 식 (2)에서 δL을 최대화하는 η는

$$\eta=\frac{1}{2}+\frac{\varepsilon^2}{(c_k-y_k)^2}$$

가 되는데 $\varepsilon=0$으로 할 때는 η값을 0.5로 하면 되고 $\varepsilon>0$일 때 η는 c_k가 y_k를 더욱더 잘 추정할수록 큰 값을 가지게 된다.

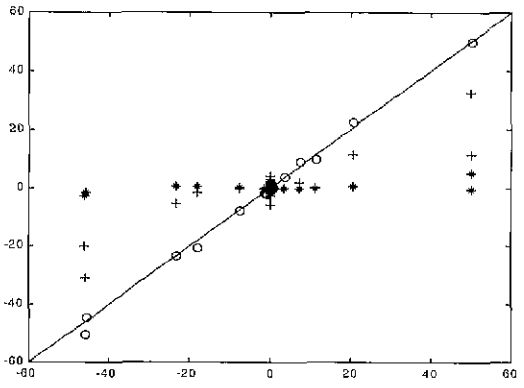
4. 실험결과

제안된 KA의 성능을 분석하기 위해 비선형 일변량 함수의 근사문제를 생각한다. 근사를 위해 두 개의 함수를 예로 들어 설명하였다. 첫 번째 함수는 Wahba & Wold [11]의 French 곡선 $f(x)=4.26(e^{-x}-4e^{-2x}+3e^{-3x})$이다. 입력값으로는 구간 [0, 3]에서 100개의 점을 등간격으로 추출하여 사용하였다. 두 번째와 세번째 함수는 함수근사를 위해 Shao[7]의 박사학위 논문에서 많이 사용되는 $f(x)=\sin^2(2\pi x)$와 다음의 불연속 다항함수이다. 입력값으로는 구간 [0, 1]에서 100개의 점을 등간격으로 추출하여 사용하였다.

$$f(x)=\begin{cases}4x^2(3-4x), & x\in[0,0.5]\\ \frac{4}{3}x(4x^2-10x+7)-\frac{3}{2}, & x\in(0.5,0.75]\\ \frac{16}{3}x(x-1)^2, & x\in(0.75,1]\end{cases}$$

함수근사를 위한 SVM에서 중요한 모수인 C, ε, σ의 선택을 위해서 VC-이론을 사용하였는데 첫 번째와 두 번째 함수의 경우에 $C=50$, $\varepsilon=0.001$, $\sigma=0.2$가 적절하였다. 그리고 세 번째 함수의 경우에는 $C=10$, $\varepsilon=0.001$, $\sigma=0.05$가 적절하였다.

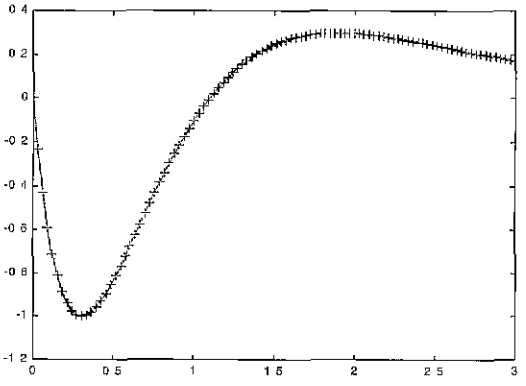
(그림 1)은 Wahba & Wold[11]의 French 곡선에 대해 QP를 사용한 SVM과 KA를 사용한 SVM이 계산한 계수 $\beta_i=\alpha_i-\alpha_i^*$ $i=1,\cdots,100$들의 값을 비교한 것인데, 여기서 *, +, o는 반복수가 각각 100회, 5,000회 그리고 100,000회 경우의 결과를 나타낸다. x-축과 y-축은 각각 QP와 KA에 의해 구해진 β_i 값들에 대한 축들이다. 반복수가 많아질수록 일치하는 경향이 있는데 이는 KA의 β_i 값이 QP의 β_i 값으로 수렴함을 의미한다.



: 100회 반복의 β_i + 5,000회 반복의 β_i o 100,000회 반복의 β_i

(그림 1) KA의 β_i값과 QP의 β_i값의 비교

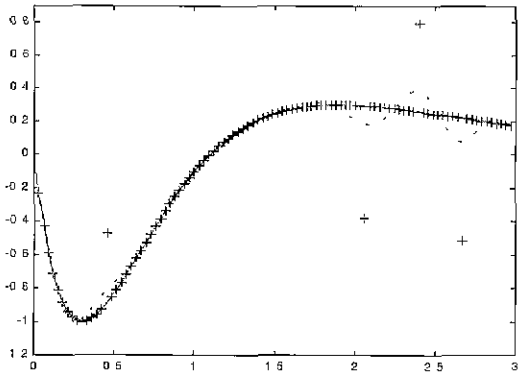
(그림 2)는 이상치가 없는 경우의 French 곡선에 대해 QP를 사용한 SVM과 KA를 사용한 SVM의 함수근사 결과를 보여준다. 여기서 +는 자료점을 나타내며 실선, 점선, 절단선(dashed line)은 각각 SVM QP, 100회 반복수의 KA, 500회 반복수의 KA에 대한 결과를 보여준다. 세 종류의 선은 거의 일치하여 구별할 수 없으며 차이가 없는 좋은 결과를 보이고 있다.



실선 SVM QP 점선 100회 반복의 KA 절단선 500회 반복의 KA

(그림 2) 이상치가 없는 French 곡선에 대한 KA와 QP의 함수근사

(그림 3)은 이상치가 있는 경우의 French 곡선에 대해 QP를 사용한 SVM과 KA를 사용한 SVM의 함수근사 결과를 보여준다. 여기서 +는 자료점을 나타내며 실선, 점선, 절단선(dashed line)은 각각 SVM QP, 100회 반복수의 KA, 500회 반복수의 KA에 대한 결과를 보여준다. 100회 반복수의 KA의 경우에는 이상치에 다소 영향을 받았지만 반복수가 500회일 때에는 이상치에 전혀 영향을 받지 않는 좋은 함수근사의 결과를 보인다. SVM QP도 이상치에 전혀 영향을 받지 않는 좋은 함수근사의 결과를 보인다. 그림에서 실선과 절단선은 거의 일치하여 구별할 수 없는데 이 사실은 SVM QP와 500회 반복수의 KA는 같은 결과를 보여줌을 설명하고 있다.



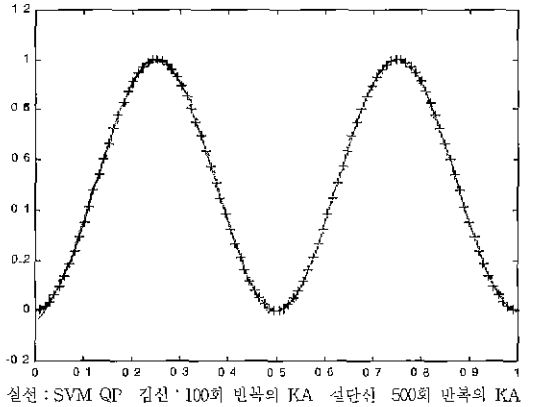
실선 SVM QP 점선 : 100회 반복의 KA 절단선 500회 반복의 KA

(그림 3) 이상치가 있는 French 곡선에 대한 KA와 QP의 함수근사

(그림 4)와 (그림 5)는 각각 이상치가 없는 경우와 있는 경우의 싸인제곱 함수에 대한 근사결과를 보여준다. 여기서 +는 자료점을 나타내며 실선, 점선, 절단선(dashed line)은 각각 SVM QP, 100회 반복수의 KA, 500회 반복수의 KA에 대한 결과를 나타낸다. 이상치가 없는 경우 세 종류의 선은 거의 일치하여 차이가 없는 좋은 결과를 보이고 있다. 그러나 이상치가 있는 경우 100회 반복수의 KA는 이상치에 아주 적게 영향을 받는 것으로 보이고, QP와 500회 반복수의 KA는 거의 일치하여 구별할 수 없는데 이 사실은 SVM QP와 500회 반복수의 KA는 같은 결과를 보여줌을 설명하고 있다 그리고 이상치에 영향을 받지 않는 좋은 결과를 보이고 있다.

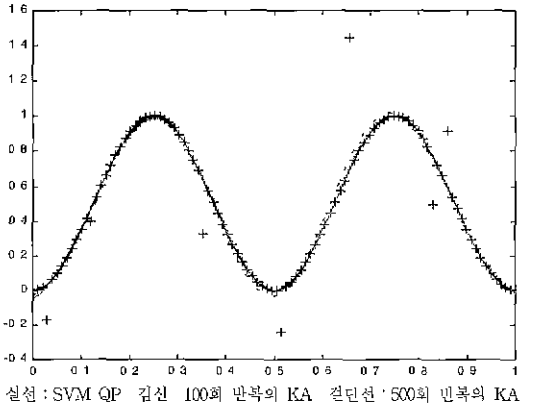
(그림 6)과 (그림 7)는 각각 이상치가 없는 경우와 있는 경우의 불연속 다항함수에 대한 근사결과를 보여준다. 여기서 +는 자료점을 나타내며 실선, 점선, 절단선(dashed line)은 각각 SVM QP, 100회 반복수의 KA, 500회 반복수의 KA에 대한 결과를 나타낸다. 이상치가 없는 경우 세 종류의 선은 불연속점인 0.5 근방을 제외한 대부분의 영역에서 대체로 좋은 결과를

보이고 있다. 한편 이상치가 있는 경우 100회 반복수의 KA는 이상치에 다소 영향을 받는 것으로 보이고, QP와 500회 반복수의 KA는 거의 일치하며 이상치에 영향을 받지 않는 좋은 결과를 보이고 있다. 이상치가 없는 경우와 마찬가지로 불연속점인 0.5 근방에서는 다소 부정확한 함수근사 결과를 보여주고 있다.



실선 : SVM QP 점선 : 100회 반복의 KA 점단선 500회 반복의 KA

(그림 4) 이상치가 없는 싸인제곱 함수에 대한 KA와 QP의 함수근사

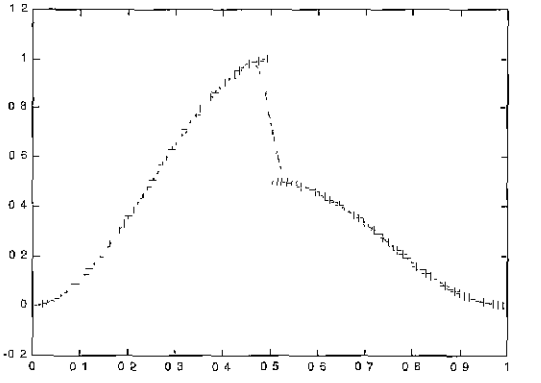


실선 : SVM QP 점선 100회 반복의 KA 점단선 : 500회 반복의 KA

(그림 5) 이상치가 있는 싸인제곱 함수에 대한 KA와 QP의 함수근사

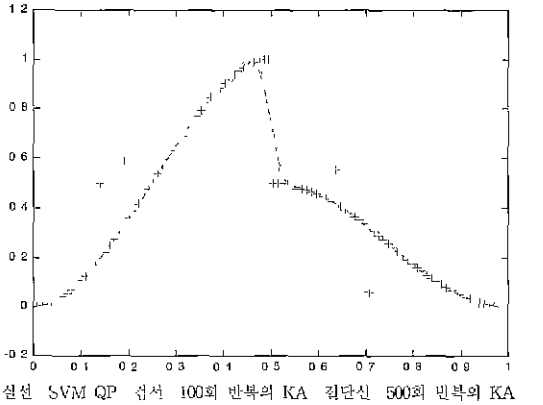
5. 결 론

모의실험은 586PC에서 MATLAB으로 구현되었다. SVM QP에 소요되는 시간은 이상치가 없는 French 곡선의 경우에는 765초, 이상치가 있는 경우에는 283초였다. 그리고 이상치가 없는 싸인제곱함수의 경우에 662초가 소요되었고 이상치가 있는 경우에는 284초가 소요되었다. 대체로 이상치가 있는 경우에 SVM QP의 소요시간이 작았다. 한편 KA의 경우는 자료의 종류에 관계없이 1,000회 반복하여 좋은 함수근사 결과를 보였는데 소요시간은 90초 정도였다.



실선 : SVM QP 점선 100회 반복의 KA 점단선 500회 반복의 KA

(그림 6) 이상치가 없는 불연속 다항함수에 대한 KA와 QP의 함수근사



실선 SVM QP 점선 100회 반복의 KA 점단선 500회 반복의 KA

(그림 7) 이상치가 있는 불연속 다항함수에 대한 KA와 QP의 함수근사

불연속 다항함수의 경우 500회 반복수의 KA는 불연속점인 0.5 근방에서 함수근사의 정확도가 QP에 비해서 떨어지는 것을 볼 수가 있으나 소요시간이 6배 내지 11배 정도 더 빨랐다. 이 사실을 고려하면 좋은 결과라고 평가할 수 있다. 그러나 1000회의 반복으로 거의 같은 결과를 얻을 수 있었다.

결론적으로 말하면 KA는 QP에 비해 적은 시간으로 거의 같은 수준의 좋은 함수근사 결과를 보인다고 말할 수 있으며, 입력값의 개수가 100개 보다 훨씬 많은

경우에는 SVM QP보다 KA가 훨씬 빠를 것으로 생각한다. 앞으로 다양한 응용분야에 적용하여 그 성능을 확인할 필요가 있다.

참 고 문 헌

[1] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," In Data Mining and Knowledge Discovery 2, Kluwer Academic Publishers, Boston, 1998.

[2] C. Campbell, and N. Cristianini, "Simple Learning Algorithms for Training Support Vector Machines," Dept. of Engineering Mathematics Technical Report, U. of Bristol, 1998

[3] N. Cristianini, C. Campbell, and J. Shawe-Taylor, "Dynamically Adapting Kernels in Support Vector Machines," NeuroCOLT2 Technical Report, NeuroCOLT, 1998

[4] T. Evgeniou, M. Pontil, and T. Poggio, "A Unified Framework for Regularization Networks and Support Vector Machines," MIT AI Laboratory, Technical Report, 1999.

[5] S Gunn, "Support Vector Machines for Classification and Regression," ISIS Technical Report, U. of Southampton, 1998.

[6] E. Osuna, R. Freund, and F. Girosi, "Support Vector Machines : Training and Applications," MIT AI Laboratory, Technical Report, 1997.

[7] X. Shao, "Model Selection Using Statistical Learning Theory," Ph. D. Thesis, U. of Minnesota, 1999.

[8] A.J. Smola, and B. Scholkopf, "A Tutorial on Support Vector Regression," NeuroCOLT2 Technical Report, NeuroCOLT, 1998.

[9] V. Vapnik, "The Nature of Statistical Learning Theory," Springer, 1995.

[10] V. Vapnik, "Statistical Learning Theory," Springer, 1998.

[11] G. Wahba, and S. Wold, "A completely automatic French curve," Communications in Statistics, 4, pp. 1-17, 1975.

석 경 하

e-mail : skh@stat.inje.ac.kr

1983년 경북대학교 사범대학 수학교육과(이학사)

1985년 서울대학교 대학원 계산통계학과(이학석사)

1990년 서울대학교 대학원 계산통계학과(이학박사)

1990년~현재 인제대학교 데이터정보학과 부교수

관심분야 : 패턴인식, 커널함수 추정

황 창 하

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

1982년 경북대학교 사범대학 수학교육과(이학사)

1984년 서울대학교 대학원 계산통계학과(이학석사)

1985년~1987년 한국통신 전임연구원

1991년 미국 미시간대학교 통계학과(이학박사)

1992년~1994년 경성대학교 전산통계학과 조교수

1995년~현재 대구가톨릭대학교 정보통계학과 부교수

관심분야 : 기계학습, 정보과학