2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

SVM을 이용한 유전자 알고리즘의 진화속도 개선 연구

*김 진 수, *손 성 한, *조 병 선, *박 강 박, **이 희 철, **장 상 근
*고려대학교 제어계측공학과, **국방과학연구소
전화 : 02-922-0863 / 핸드폰 : 011-9828-2191

A Study of Accelerated Evolution Speed of Genetic Algorithm using SVM

Jinsu Kim, Sung-Han Son, Byungsun Cho, Kang-Bak Park,
Hee-Churl Lee, Sang-Geun Jang
Dept. of Control and Instrumentation Engineering, Korea University
E-mail : jinux73@korea.ac.kr

Abstract

The chromosomes of Genetic Algorithm(GA) are classified to be good or not to be by Support vector machines(SVM), and then the only good chromosomes are adopted to the evolution process. By this way, computational load becomes low, so the evolution speed of Genetic Algorithm modified by SVM can be much accelerated than the conventional GA.

I. 서론

본 논문에서는 SVM(Support Vector Machines)을 이용하여 연산속도를 높인 유전자 알고리즘(Genetic Algorithm) 기법을 기본 개념으로 하여 연구를 수행하고자 한다. 지난 수십 년 동안 자연현상에 기초한 알고리즘들에 대하여 많은 관심들이 집중되어왔다. 이를 바탕으로 생긴 이론들을 보면, 이른바 나비 효과로 대변되는 '카오스(Chaos)', 신경망을 모사한 '신경회로망(Neural Network)', 언어의 불확실성에 바탕을 둔 '퍼지시스템(Fuzzy systems)', 등과 더불어 유전자의 교배 및 돌연변이 등을 모델화한 '유전자 알고리즘(Genetic Algorithm)'이 있다.

그 중에서도 유전자 알고리즘은 최적화 문제에 좋은 솔루션을 제공한다는 것이 실험적으로 알려져 오고 있다. 유전자 알고리즘이 제안된 당시에는 컴퓨터의 연산능력이 우수하지 못하였기 때문에, 신경회로망이 초기에 그러했던 것처럼 많은 관심을 끌지 못했었다. 그러나 병렬 컴퓨터들이 등장하고, 개인용 컴퓨터들의 연산속도가 향상되면서 점차 관심이 높아졌으며, 현재는 어느 정도 완숙기에 접어들었다고 할 수 있다. 또한 유전자 알고리즘은 일반적으로 이론적 배경 부족과 진화 유전자의 개수 증가에 따른 계산량이 많다는 단점은 있으나, 그 실험적 결과들은 기대 이상이다. 특히 off-line 최적화 문제에 있어서 좋은 성능을 보이는 방법이다.

본 연구에서 처음으로 SVM 알고리즘을 이용하여 진화과정의 단계마다 적합도(fitness)에 따라 계속적으로 진화에 참여하는 유전자(chromosome)와 도태되는 유전자(chromosome)를 분석하여 진화 및 도태의 원인을 찾아내고, 추후 계속되어 진행되는 진화과정에서 이를 이용하여 진화속도가 빨라지도록 하였다.

II. 서포트 벡터 학습방법 (Support Vector Machines)

2.1 서포트 벡터 학습방법

서포트 벡터 학습 방법은 최근에 패턴 분류 및 함수 근사 등의 문제에서 매우 우수한 성능을 보이고 있는 새로운 학습 방법이며, 특히 화상 정보의 압축에도 많이 이용되고 있는 알고리즘이다. 이 방법은 패턴 및 통계적 처리가 필요한 정보를 인식한 후 분류가 가능한 영역으로 분류하고, 각 영역별로 최적화된 압축정보로 영역의 대표값을 정하는 방법이다. 유전자 알고리즘에 서포트 백터 학습 방법의 통계적 분류방법을 적용시켜 진화과정의 단계마다 적합도(fitness)에 따라 계속적으로 진화에 참여하는 유전자와 도태되는 유전자를 분석하여 진화 및 도태의 원인을 찾아내고, 추후 계속되어 진행되는 진화과정에서 이를 이용하여 계산 시간 및 적합도가 높은 결과가 도출되도록 하였다.

본 논문에서 소개하는 서포트 벡터 학습 방법은 한 개의 은닉 층을(hidden-layer) 갖는 MLP (multi-layer perception) 및 RBF(radial basis function) 신경망 등을 대상으로 하여 다음과 같은 장점을 갖는 해를 제공하는 특징을 가진다.

- 은닉 노드의 개수가 자동으로 결정될 수 있다.
- 기울기 강하 기법 (gradient descent method) 등의 학습 방법론이 가지고 있었던 지역적 최적해(local optimum)로 수렴하는 문제가 없어서 반드시 성능지수(이진 분류 문제에서는 maximal margin, 함수 근사 문제에서는 ε-insensitive error로 표현되는 지수)를 전 역적으로 최소화할 수 있는 전 역적 최적해(global optimum)를 찾을 수 있다.
- 유도과정이 통계적 학습이론(statistical learning theory)으로 설명될 수 있기 때문에, PAC 상한(probably approximately correct bound)을 만족시키는 의미로 충분한 일반화 능력(generalization capability)이 확보된다.
- 서포트 벡터(support vector)는 트레이닝 데이터(training data)의 부분집합 (small subset)으로 구성된다.

2.2 최적분리면(Optimal separating hyper-plane)

두개의 클래스를 갖는 패턴 분류 문제의 목적은 두 개의 클래스를 구분 지을 수 있는 분류기(classifiers)를 찾는데 있다. 학습하고자 하는 데이터가

$$(x_1, y_1),\ (x_2, y_2),\ \cdots,\ (x_m, y_m) \qquad (1)$$

과 같이 주어진다고 하자. 이때, m은 데이터의 개수를 의미하고, $x \in R^n$는 입력 데이터, $y \in \{-1, +1\}$는 출력 데이터이다. 학습 데이터는 $y=+1$인 부분집합과 $y=-1$인 부분집합으로 이루어져 있는데 이 두 부분집합은 다음과 같은 초평면(hyper-plane)에 의해서 구분되어질 수 있다.

$$\langle w, x\rangle + b = 0 \qquad (2)$$

이때, $w \in R^n$와 $b \in R$는 초평면을 이루는 하중벡터(weight vector)와 바이어스(bias)항 이며, 다음의 부등식을 만족한다.

$$\begin{aligned} \langle w, x_i\rangle + b \geq 1 \ \text{ for } y_i = 1 \\ \langle w, x_i\rangle + b \leq -1 \ \text{ for } y_i = -1 \end{aligned} \qquad (3)$$

그리고, 위 식은

$$y_i(\langle w, x_i\rangle + b) \geq 1, \quad i = 1, \cdots, m \qquad (4)$$

과 같이 표현된다.

식 (2)를 만족하는 초평면은 무수히 많이 존재하는데 그 중에서 최적의 초평면을 찾기 위하여 초평면과 가장 가까운 데이터와 초평면과의 거리를 마진(margin)이라 정의한다. 이 마진이 최대가 되는 초평면을 최적 분리면(optimal separating hyper-plane)이라 하고, 서포트 벡터 학습 알고리즘은 이러한 최적 분리 면의 w와 b을 찾는데 그 목적이 있다. 마진이 최대가 되는 최적 분리 면을 찾는 문제는 다음의 최적화 문제를 푸는 것으로 표현된다.

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 \\ &s.t. \ \ y_i(\langle w, x_i\rangle + b) \geq 1, \quad i = 1, \cdots, m \end{aligned} \qquad (5)$$

식 (5)는 에러 없이 분류가 되는 경우이다. 그러나, 에러 없이 분류가 불가능한 경우나, 어느 정도 에러의 포함을 필요로 하는 경우에 대해서 에러를 포함하는 분류를 고려한다. 새로운 변수(slack-variable) $\xi_i \geq 0$를 추가하면 최적화 문제의 제약조건은

$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \cdots, m \qquad (6)$$

이 되어, 에러를 포함하게된다. 그러므로, 식 (5)의 최적화 문제는 다음과 같이 바뀌게 된다:

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ &s.t. \ \ y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, \quad i = 1, \cdots, m. \end{aligned} \qquad (7)$$

여기서 C는 조정상수(regularization constant)로 $C>0$인 값 중에서 선택한다.

III. SVM에 의해 수정된 유전자 알고리즘 (SVMGA)

3.1 유전자 알고리즘(GA)

자연계에 있는 생물의 진화과정에 있어서, 어떤 세대(generation)를 형성하는 개체(individual)들의 집합, 즉 개체군(population) 중에서 환경에 대한 적합도(fitness)가 높은 개체가 높은 확률로 살아남아 재생

(reproduction)할 수 있게 되며, 이때 교배(crossover) 및 돌연변이(mutation)로서 다음 세대의 개체군을 형성하게 된다. GA에서 개체의 수를 개체군의 크기(population size)라고 한다. 각각의 개체는 염색체를 가지고 있으며 염색체는 복수개의 유전자(gene)의 집합으로 구성된다. 유전자의 위치를 유전자좌(locus)라 하고 유전자가 취하게 되는 유전자의 후보를 대립 유전자(형질, allele)라고 한다. 생물의 경우 염색체는 어떤 개체의 특징을 상세하게 결정하게 되는데 예를 들어 머리가 검은 것은 염색체 중에 이러한 특징을 나타내도록 하는 유전자의 조합이 존재하기 때문이다. 이와 같이 유전자에 의해 결정되는 개체의 형질을 표현형(phenotype)이라고 하고 이에 대응되는 염색체의 구조를 유전형(genotype)이라 한다. 여기에서 표현형이 여러 개의 유전자좌의 영향을 받아 복잡한 형태가 결정되는데 이것을 에피스타시스(epistasis)라고 한다. 또한 표현형을 유전형으로 바꾸는 것을 코드화(coding) 그 역을 디코드화(decoding)라고 한다. GA는 이와 같이 생물의 진화과정을 인공적으로 모델링 한 알고리즘이다.

3.2 SVMGA

먼저, SVMGA의 알고리즘을 살펴보게되면 아래와 같다.

①Evaluate initial population

Generational loop

②Assign fitness-value to entire population

③Generate training data and do SVM

④Select individuals for breeding

⑤Recombine selected individuals (crossover)

⑥Perform mutation on offspring

⑦Adopt the SVM criterion

(classify GOOD and BAD individuals)

⑧Evaluate offspring, call objective function

⑨Reinsert offspring into current population

loop end.

여기서, 스텝 3과 7를 제외하면 기존의 GA가 된다.

스텝 2에서 생성된 데이터가 SVM의 트레이닝 데이터(training data)가 된다. 즉 식(1)에서 x_i는 유전자값이고 y_i는 적합도(fitness)가 된다. 스텝 7에서 생성된 SVM의 분류기(classifier)는 생성된 유전자가 시스템에 들어가 적합도를 생성할 것인지 아닌지를 결정하게 된다. 결국, 분류기로 걸러진 도태되는 유전자(-클래스)는 시스템에 부하를 걸어주지 않게 되고 따라서 분류기로 걸러진 양질의 유전자(+클래스)만이 진화에 참여하게 되서 진화속도가 빨라질 수 있게 된다.

Ⅳ. 예제와 시뮬레이션 결과

본 장에서는, 아래와 같은 식을 최소화하는 De Jong[1]의 확장된 예제에 적용해 기존 GA와 SVM으로 수정된 GA(SVMGA)를 비교 검토하였다.

$$\min f_1(x) = \sum_{i=1}^{n} x_i^2, \quad -512 \le x_i \le 512 \qquad (8)$$

여기서 n=2이고, 개체의 수(NIND)는 100으로, generation gap(GGAP)은 0.9로 하였다. 즉, 진화를 통해 새롭게 생성되는 유전자는 90이다. 그중에서 어느 정도가 SVM 분류기로 걸러지는지, 걸러진 양질의 유전자가 정해놓은 기준에 부합하는지 아래의 시뮬레이션 결과를 통해 알아본다

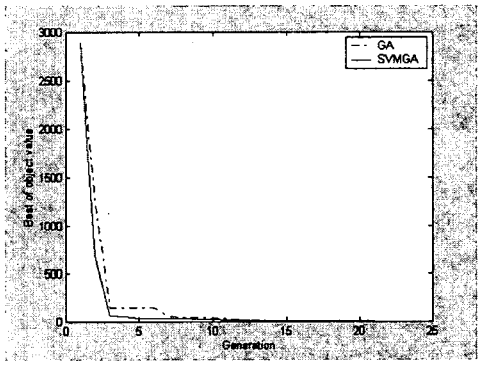

그림 1 Generation에 따른 Object value의 최소값

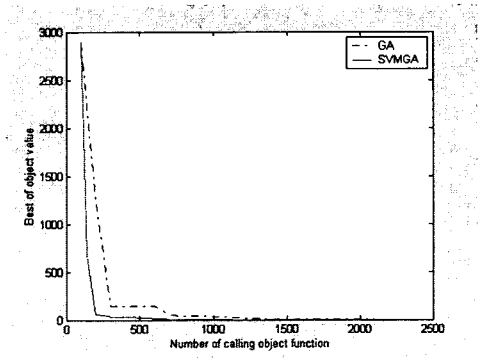

그림 2 시스템부하에 따른 Object value의 최소값

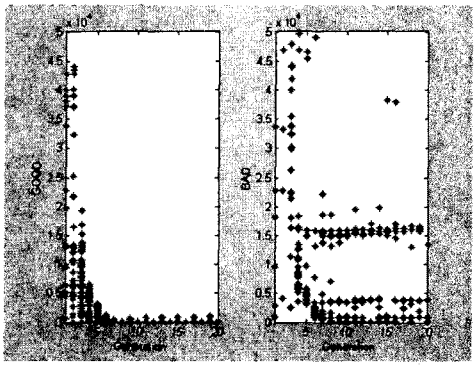

그림 3 분류기로 걸러진 유전자의 분포

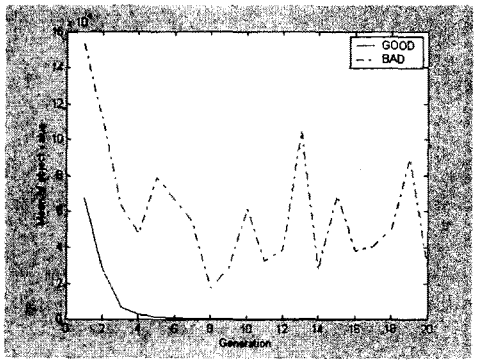

그림 4 분류기로 걸러진 유전자의 평균

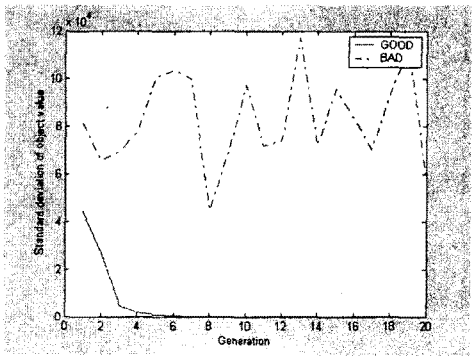

그림 5 분류기로 걸러진 유전자의 분산

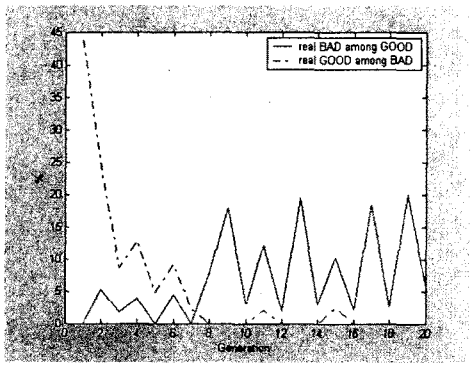

그림 6 분류기의 에러율

V. 결론

지금까지 SVM 분류기로 걸러진 양질의 유전자가 도태되는 유전자보다 시스템값(object value)의 평균, 분산, 분포가 좋음을 알 수 있었다. 즉 분류기로 걸러진 양질의 유전자만으로도 충분히 좋은 성능을 낼 수 있음을 보여주는 것이고 결과적으로 적은 데이터 처리로 인해 제안하는 알고리즘의 진화속도가 기존의 방법보다 빨라지는 경향을 보인다고 볼 수 있다.

이 연구는 국방과학연구소(UD020002ED)에 의해서 지원받았습니다

참고문헌

[1] K. A. De Jong, "Analysis of the Behaviour of a Class of Genetic Adaptive Systems," PhD Thesis, University of Michigan, Ann Arbor, 1975.

[2] T. Back, H.P. Schwefel, "An overview of evolutionary algorithms for parameter optimization, Evolutionary Computation, Vol. 1, No. 1, pp. 1-23, 1993.

[3] A. J. Chipperfield and P. J. Fleming, "The Matlab Genetic Algorithm Toolbox"

[4] Steve Gunn, "Support Vector Machines for Classification and Regression," University of Southampton, 1998, ISIS Technical Report

[5] Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge UK, Cambrrdge Univ. Press, 2000.