다수 유전자 프로그램의 최적 결합을 위한 확률적 탐색 방법

정 제 균° 장 병 탁
서울대학교 컴퓨터공학과
E-mail: {jgjoung, btzhang}@scai.snu.ac.kr

A Probabilistic Search Method for Optimal Combination of Multiple Genetic Programs

Je-Gun Joung° Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

유전자 프로그래밍은 고정적인 구조가 아닌 가변 길이의 트리 구조를 가지고 있어서 여러 세대를 통하여 다양한 개체들을 만들어 낸다. 이러한 특징은 위원회 머신(committee machines)을 구축하는데 있어서 자연스럽고 또한 효과적인 알고리즘일 수 있다. 하지만 해결해야할 요소 중 하나는 다수의 개체들에서 결합할 개체의 선택과 개체의 수를 결정하기 위한 방법이다. 본 논문에서는 효과적인 개체들의 결합이 되기 위한 새로운 탐색방법을 소개한다. 이 방법은 확률적인 진화 탐색을 바탕으로 하고 있다. 제안된 방법을 여러 가지 분류 문제에 적용하였으며 실험을 통하여 탐색의 특성과 일반화 성능을 분석하였다.

1. 서 론

위원회 머신(committee machines)은 여러 모델들의 의사를 결합함으로써 일반화 성능을 향상시키는 시스템이다. 여기서 모델들은 신경망(neural network), 결정 트리(decision tree) 또는 다른 학습 모델들을 포함한다. 최근 몇몇 연구들은 진화 알고리즘으로 모델을 생성하고, 생성된 모델들을 결합한 위원회 머신들을 제시하고 있다[1][2][3][4]. 여기서의 모델은 하나의 개체를 의미한다. 이러한 배경은 진화 알고리즘이 개체 집단을 유지하면서 많은 세대 수를 통하여 다양한 개체들을 생성하는 특징을 가지고 있다. 이는 위원회 머신의 구축을 위한 다양한 자원을 제공할 수 있는 장점을 가지고 있다.

진화 알고리즘을 통한 위원회 머신에 대한 연구들의 대부분은 결합 방법 또는 다양한 개체의 생성에 초점을 두고 있다. 가장 일반적인 결합 방법으로는 가중치에 의한 평균(weighted average), 과반수 투표(majority voting)이다. 다양한 모델의 생성은 개체의 구조 뿐만 아니라 파라미터에 관한 측면이다. 이와는 다른 관점으로써, 생성된 다양한 개체들로부터 어떤 개체들로 위원회를 구성할 지의 여부가 하나의 관심거리가 된다. Yao는 최근에 이를 위하여 유전자 알고리즘을 사용하여 위원회 구성원을 위한 최적의 서브집합을 발견하는 방법에 대하여 부분적으로 언급하였다[2].

본 논문에서는 위원회의 구성원 수를 조정함으로써 최적의 위원회를 구성하는 방법을 제시한다. 이러한 방법은 포아송 분포를 근본으로 하고 있고 위원회 수의 랜덤 분포에 대한 가정에서 출발한다. 이러한 알고리즘을 ADIMOC (Adaptive DIscovery Method for Organizing Committees)이라고 명명하였다. ADIMOC의 장점은 위원회 구성을 위한 모든 가능한 조합을 고려하지 않고 빠른 탐색을 할 수 있다는 것이다.

논문의 구성은 다음과 같다. 2장에서는 위원회 머신의 기본 개념과 진화 알고리즘에서 접근한 위원회 머신에 대하여 기술한다. 3장에서는 위원회 구축을 위한 단계로써 전체적으로 개체들의 진화와 위원회의 진화를 구분하여 기술한다. 그리고 4장에서는 분류 문제에 대한 실험 결과를 보인다. 마지막 5장에서는 결론을 제시한다.

2. 진화 알고리즘에 대한 위원회 머신

일반적으로 위원회 머신의 목적은 하나의 결정보다 우수한 전체적인 결정을 향하여 전문가 또는 위원회의 구성원들에 의해 얻어진 지식(knowledge)들을 융합시키는 것이다[5]. 위원회 머신은 앙상블 평균(ensemble averaging), 부스팅(boosting), 전문가들의 혼합(mixture of experts)등을 포함한다. 그들은 구조에 따라 두 부류로 나뉘어 질 수 있다. 하나는 앙상블 평균, 부스팅을 포함하고 있는 정적구조를 가진 앙상블 방법들이다. 이 방법에서는 입력값들이 결합하고자 하는 모델에 직접적인 개입을 하지 않는다. 다른 하나는 전문가들의 혼합과 같은 동적구조를 가진 지역적 전문가(local expert)들의 결합방법이다. 여기서는 입력값이 전문가들의 의사 결정에 직접적인 개입을 한다.

위원회 머신을 위한 대부분의 연구들이 단순히 신경망이나 결정트리 또는 확률모델의 결합[6][7][8]을 시도하고 있는 반면, 진화적 기법의 적용은 극히 일부이다[1][2][4]. 일반적으로 모델 생성을 위한 진화적 학습에서는 진화를 시킨 다음 가장 좋은 모델만을 선택하고 나머지는 버린다. 그것은 진화에 투자했던 계산적인 시간과 메모리의 낭비이다. 이러한 결점은 위원회를 형성하기 위하여 다수의 개체들을 결합함으로써 해결될 수 있다. 이러한 방법의 특징은 적은 노력으로 새로운 데이터에 대하여 예측의 정확성을 향상시킬 수 있다는 것이다. 대부분의 진화알고리즘을 적용한 위원회 머신들은 정적구조를 가진다. 진화가 종료가 되고 다수의 개체들은 입력값의 개입이 없이 그들의 출력들의 결합으로 최종적인 값을 보인다.

3. 위원회의 형성

3.1 개체들과 위원회의 진화적 탐색

제안된 알고리즘에서는 전체적으로 개체들의 진화와 위원회의 진화의 두 단계로 나뉘어 질 수 있다.

단계 1: 개체의 진화
- 개체들의 생성과 진화
- 다양하고 높은 적합도(fitness)를 가진 개체들을 저장

단계 2: 위원회의 진화
- 저장된 개체들로부터 후보집합을 생성
- 후보집합에서 최적의 위원회를 탐색

첫 번째 단계의 목적은 위원회 구성을 위한 자원을 제공하는 것이다. 먼저 각각의 개체를 A_i 라고 표기한다. 그러면 각 세대의 개체집단은 $A(g)=\{A_i(g)\}_{i=1}^{M}$으로 정의될 수 있다. 그리고 G 세대수 동안 전체 개체들의 공간을 $A=\{A(0),...,A(G)\}$로 나타낸다. 진화를 하는 동안 개체 A_i의 적합도는

$$F(A_i)=E(A_i)+\alpha C(A_i) \tag{1}$$

로 정의된다. 여기서 $E(A_i)$는 개체 A_i의 에러를 나타내고, $C(A_i)$는 복잡도를 나타낸다. 복잡도는 노드의 수와 깊이를 포함한다. α는 에러와 복잡도의 균형을 유지하기 위한 오캄 인자(ocaam factor)이다[9].

다음 단계에서는 위원회의 탐색을 목적으로 한다. i번째 위원회를 V_i로 나타내고, 진화알고리즘에 있어서 개체로써 고려된다. 각각의 위원들은 k개의 구성원 v 를 포함하고 있다.

$$V_i=(v_1,...,v_k), \quad v\in A \tag{2}$$

각각의 v는 개체의 진화단계에서 생성된 개체들 중 하나이다. 위원회의 적합도 역시 복잡도 항목을 추가하여 다음과 같이 나타낼 수 있다.

$$R(V_i)=E(V_i)+\alpha C(V_i) \tag{3}$$

여기서 $E(V_i)$는 개체들의 결합에 의한 출력과 목적값과의 차이에 대한 에러이다. $E(V_i)$는 다음 식에 의해 계산된다.

$$E(V_i)=\sum_{t=1}^{N}\left(\sum_{j=1}^{k} w_j v_j(\mathrm{x}_t)-y_t\right) \tag{4}$$

t번째 입력 벡터 x_t에 대한 구성원의 출력 $v_j(\mathrm{x}_t)$에 대하여 가중치 w를 곱해서 나온 출력값들의 합과 목표값 y_t와의 차이로써 에러를 계산한다. 본 논문에서는 w를 결정하는 알고리즘으로 일반화된 앙상블 방법을 사용한다[10]. 다음 항목인 복잡도는 위원회 수를 고려하여 $C(V_i)=(k/N)$로써 나타낸다.

많은 세대수에 걸쳐 생성된 개체들의 수를 모두 위원회 구성을 위한 대상으로 고려할 수 없기 때문에 적합도가 좋은 몇몇 개체만을 고려한다. 이들 개체를 후보 집합 Q라고 하고, Q에 속하는 후보 Q_g는 각 세대에서 가장 높은 적합도를 가진 개체이다.

$$Q=\{Q_g\}_{g=1}^{S}, \quad Q_g=A_{best}(g) \tag{5}$$

여기서 S는 후보집합의 크기이다. 후보 집합에서 Q_i가 위원회의 구성원으로 선택될 확률은 다음과 같다.

$$P(Q_i)=\frac{\exp(\gamma R(Q_i))}{\sum_{j=1}^{S}\exp(\gamma R(Q_j))} \tag{6}$$

여기서 γ는 상대적인 적합도 차이를 조절하는 상수이다. 식 (6)은 위원회 구성을 위하여 적합도가 낮은 후보 보다 높은 후보가 선택될 확률을 높여 준다.

3.2 확률적 방법에 의한 위원회 탐색

최종목적은 최적의 위원회 V^*를 찾는 것이다. 가능한 탐색공간은 (2^S-1)로 상당히 큰 공간이다. 이러한 부담을 감소시키기 위하여 위원회의 수에 대한 분포를 통하여 탐색을 한다. 일단 데이터 D를 관찰하였을 때 위원회 $V_i(g)$의 사후분포(posterior distribution)는 다음과 같다.

$$P(V_i(g)|D)=\frac{P(D|V_i(g))P(V_i(g))}{P(D)} \tag{7}$$

각각의 위원회는 구성원의 수 k만큼 구성원들을 포함하고 있고 k는 포아송 분포(poisson distribution)에 따른다.

$$\begin{aligned} &P(D|V_i(g),k,\lambda)P(V_i(g),k,\lambda)\\ &=P(D|V_i(g),k,\lambda)P(V_i(g)|k,\lambda)P(k|\lambda)P(\lambda)\\ &=P(D|V_i(g),k,\lambda)P(k|\lambda)P(\lambda) \end{aligned} \tag{8}$$

여기서 $P(D|V_i(g),k,\lambda)$은 에러 항목이고, $P(V_i(g)|k,\lambda)$은 균일한 분포를 가지므로 제거할 수 있다. 그리고 $P(k|\lambda)$는 포아송 분포를 나타내고 있고, 포아송 분포의 평균값 λ에 따라 램덤한 k를 발생시킨다. λ의 확률은 다음과 같다.

$$P(\lambda)=\frac{F(V_{best}^{\lambda}(g-1))}{\sum_{i=2}^{K}F(V_{best}^{i}(g-1))} \tag{9}$$

여기서 K는 최대 가능한 구성원의 수이다. 식 (9)는 세대에서 가장 높은 적합도를 가지는 위원회의 구성원 수를 다음 세대의 λ로 설정할 확률을 의미한다. 그래서 그 근처에서 랜덤하게 발생한 구성원 수에 따라 중점적인 탐색을 하게된다.

4. 실험 및 결과

실험으로 UCI 데이터중 4개의 분류문제를 적용하였다. 먼저, 실험에 사용된 유전자 프로그래밍 매개 변수는 다음과 같다. 개체 집합의 수는 200, 총 세대수는 100, 선택 방법으로는 랭킹선택(ranking selection)을 이용하였다. 돌연변이율은 0.01이고 함수집합은 $\{+,-,\times,\div,\geq\}$ 으로써 산술연산자와 비교연산자를 포함하고 있다. 다음, 위원회의 진화에 사용된 매개 변수는 다음과 같다. 개체집합의 수는 25, 총 진화횟수는 50, 최대 후보 집합 크기는100, 최대 구성원의 수는 20이고 γ는 100이다. 그리고 포

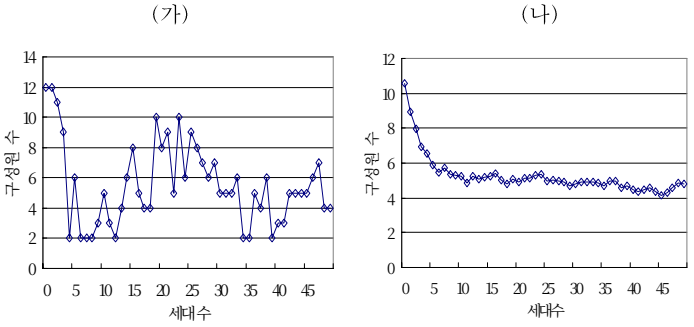

그림 1. (가) 세대수가 감에 따른 포아송 분포의 평균값의 변동 (나) 포아송 분포의 평균값에 대하여 100번 시뮬레이션한 평균값 아송 분포의 평균 λ의 초기값을 10으로 설정하였다.

그림 1은 세대가 감에 따른 포아송 분포의 평균값 λ의 변화를 보이고 있다. 왼쪽 그림은 λ의 변화 경향의 예를 보이기 위하여 한번 시뮬레이션의 결과이다. 오른쪽 그림에서 최종 세대에서 λ는 4로 근접한다.

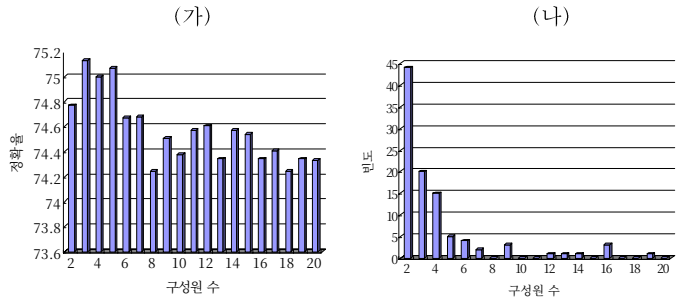

그림 2. (a) 고정된 구성원의 수에 따른 정확율 (b) 100번 시뮬레이션 결과로써 ADIMOC을 이용하여 탐색한 위원회의 구성원 수에 따른 빈도

		평균 에러율		
데이터 집합	데이터 수	C4.5	일반적인 GP	ADIMOC
Breast	699	5.86±3.32	3.69±2.2	3.55±2.1
Credit card	690	14.03±3.28	14.71±4.1	14.34±4.0
Heart	270	24.05±7.24	19.25±7.7	19.03±7.7
Diabetes	768	22.93±3.98	25.30±5.2	25.14±5.3

표 1. C4.5, 일반적인 유전자 프로그래밍과 ADIMOC을 이용한 유전자 프로그래밍 대한 분류의 에러율

그림 2의 왼쪽 그림에서 2에서 5까지의 구성원수를 가질 때 정확도가 높을 결과를 보이고 있다. 오른쪽 분포표는 100번 시뮬레이션 결과 역시 2에서 5까지의 구성원수가 많은 빈도수를 가지는 것으로 나타났다. 따라서 ADIMOC이 적합한 구성원의 수를 잘 탐색한 것으로 검증된다. 표1은 ADIMOC에 의한 방법이 기존의 알고리즘의 일반화 성능을 개선시켰음을 보여준다.

5. 결 론

본 논문에서는 최적의 위원회를 탐색하기 위한 새로운 방법을 제시하였고 UCI repository 데이터에 대한 실험 결과를 보이고 있다. 제안된 방법은 확률적 진화 기법을 이용하여 구성원 수를 능동적으로 조정함으로써 탐색하는 특징을 가지고 있다. 그리고 복잡도 함수를 적용하여 비교적 계산 시간이 감소되도록 적은 수가 결합되도록 유도하였다. 실험 결과는 이들 경향을 잘 뒷받침하고 있다.

향후 연구로는 보다 다양한 개체를 포함하는 후보집합을 생성하는 방법과 유니모달(unimodal) 분포 대신 멀티모달(multimodal) 분포를 통하여 위원회를 탐색하는 방법에 중점을 두고자 한다.

감사의 글: 본 연구는 한국과학재단 핵심전문연구(과제번호 981-0920-107-2)와 과학기술부 뇌연구개발사업(BR-2-1-G-06)에 의하여 일부 지원되었음.

참고문헌

[1] Zhang, B.-T. and Joung, J.-G., "Time series prediction using committee machines of evolutionary neural trees," *Proceedings of the Congress on Evolutionary Computation*, Vol. 1, pp. 281-286, 1999.

[2] Yao, X. and Liu, Y., "Making use of population information in evolutionary artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, 28B(2), pp. 417-425, 1998.

[3] Cho, S. B., "Combining modular neural networks developed by evolutionary algorithm," *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 647-650, 1997.

[4] Zhang, B.-T. and Joung, J.-G., "Enhancing robustness of genetic programming at the species level," *Genetic Programming Conference* (GP-97), Morgan Kaufmann, pp. 336-342, 1997.

[5] Haykin, S., *Neural Networks, a Comprehensive Foundation*, Prentice Hall, 1994.

[6] Opitz, W. and Shavlik, J. W., "Actively searching for an effective neural-network ensemble," *Connection Science*, pp. 337-353, 1996.

[7] Hashem, S., "Optimal linear combinations of neural networks," *Neural Networks*, 10(4), pp. 599-614, 1997.

[8] Lemm, J. C., "Mixtures of gaussian process priors," *The NinthInternational Conference on Artificial Neural Networks (ICANN 99)*, 1999.

[9] Zhang, B.-T., Ohm, P. and Müehlenbein, H., "Evolutionary induction of sparse neural trees," *Evolutionary Computation*, 5(2), pp. 213-236, 1997.

[10] Perron, M. P. and Cooper, L. N., "When network for function interpolation," *Neural Computation*, Vol. 3, pp. 213-225, 1991.

[11] Murphy, P. M. and Aha, D. W., UCI Repository of Machine Learning Datasets (machine-readable data repository), University of California-Irvine, *Department of Information and Computer Science*, 1994.