

연속 변수 함수 최적화를 위한 탐색점 분포 학습 알고리즘

신수용 장병탁
서울대학교 컴퓨터공학부
{syshin, btzhang}@scai.scai.ac.kr

Estimation of Distribution Algorithm for Continuous Function Optimization

Soo-Yong Shin Byoung-Tak Zhang
Dept. of Computer Science and Engineering,
Seoul National University

요 약

기존의 진화 연산의 한계를 극복하기 위해서 탐색점 분포 학습 알고리즘(Estimation of Distribution Algorithm)이 부각되고 있다. 탐색점 분포 학습 알고리즘은 데이터의 분포를 파악하고, 파악된 분포를 이용해서 새로운 학습 데이터를 생성하는 일련의 과정을 통하여 최적화 문제를 해결하는 방법이다. 그런데, 기존의 탐색점 분포 학습 알고리즘들은 대부분 이진 벡터값을 가지는 최적화 문제들만을 대상으로 하고 있다. 본 논문에서는 비감독 확률 신경망 모델인 헬름홀츠 머신을 이용해서 데이터의 분포를 학습하여 연속 함수 최적화 문제를 해결하는 방법을 개발하였다. 테스트 함수들에 대해서 실수 표현형을 사용한 유전자 알고리즘과 결과를 비교하여 제안하는 방법의 우수성을 검증하였다.

1. 서 론

기존의 진화 연산들은 빌딩 블록 개념의 한계와 교차 연산의 특징으로 인해서 변수들간의 의존성이 있는 최적화 문제들은 쉽게 해결하지 못하는 한계를 가지고 있다. 이러한 단점을 극복하기 위해서 여러 가지 방법들이 제안되고 있으나 그 중 탐색점 분포 학습 알고리즘(Estimation of Distribution Algorithm, EDA)이 대표적이라고 할 수 있다[1, 2]. 탐색점 분포 학습 알고리즘은 데이터를 학습해 데이터의 분포를 표현할 수 있는 모델을 형성하고, 그 모델로 새로운 데이터를 생성한 후, 생성된 데이터 중에서 적합한 것을 선택하고, 선택된 데이터로 모델을 조금씩 수정해 나가는 방법이다. 기존의 진화 연산과 다른 점은 교차 연산 또는 돌연변이 연산 등을 사용하지 않고, 단지 선택 연산만을 사용하면서 데이터의 분포를 파악하는 방법으로 최적해를 발견한다는 것이다.

지금까지 많은 연구가 수행되었는데 모델의 표현 능력에 따라 4가지 정도로 구분할 수 있다.[1, 2]. 첫 번째 방법은 단순하게 생각하여 의존성이 없다고 가정하고 문제를 해결하는 방법이고, 두 번째 방법은 쌍의존성(pairwise dependency)을 고려하는 방법이다. 세 번째 방법은 트리 형태를 이용하는 것이고, 마지막 방법은 네트워크 형태로 의존성을 표현하는 방법인데 가장 복잡한 형태의 의존성을 표현할 수 있다.

그런데, 대부분의 연구는 이진 벡터 함수들의 최적값을 구하는데 그치고 있었다. 그러나 실제 문제들은 대부분의 경우

연속 함수로써 실수 벡터값을 가지는 문제들이다. 따라서 최근 들어 실수 벡터값들을 가지는 함수들에 대한 연구가 진행되고 있으나 아직 뚜렷한 결과를 보여주지 못하고 있다 [3].

본 저자들은 이전의 연구를 통하여 헬름홀츠 머신을 이용해서 이진 벡터값을 가지는 디셉티브(deceptive) 함수 최적화 문제를 성공적으로 해결하였다[4]. 본 논문에서는 기존의 연구를 바탕으로 하여 이진 벡터값이 아닌 실수 벡터값들을 가지는 최적화 함수들을 해결할 수 있도록 확장하였다. 다음절에서 헬름홀츠 머신에 대해서 설명하고, 3절에서 구체적인 알고리즘을 소개한 후, 4절에서 여러 가지 실험 및 결과를 보이고, 5절에서 결론을 내리고자 한다.

2. 헬름홀츠 머신

헬름홀츠 머신(Helmholtz machine, HM)은 대표적인 생성 모델 중 하나로 인식 모델(recognition model)과 생성 모델(generative model)을 결합하여 만든 확률신경망이다[5]. 기본적인 헬름홀츠 머신은 다수의 유닛으로 이루어지는 2개 이상의 층으로 구성된다. 각 층의 유닛은 뉴런과 유사한 확률 유닛이고, 서로 다른 층의 유닛들은 인식 모델과 생성 모델을 형성하는 각각 서로 다른 2개의 가중치에 의해서 계층적으로 연결된다. 그림 1에 2개의 층을 가지는 헬름홀츠 머신을 소개하고 있다. 하위 층은 입력을 받는 입력 노드(visible variable)로 구성되어 있고, 상위 층은 은닉 노드(latent variable)로 구성되어 있다. 인식 연결(recognition connection)은 점선으로 표시되어 있으며, 생성 연결(generative connection)은

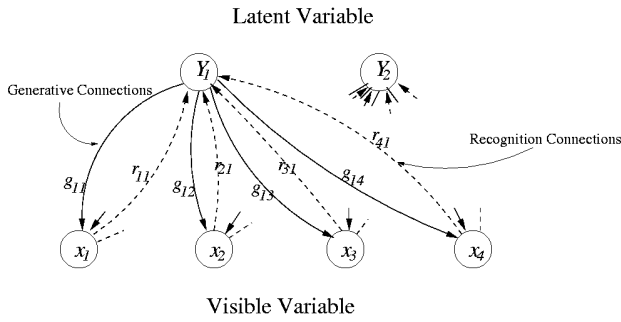


그림 1. 헬름홀츠 머신

실선으로 나타나 있다.

헬름홀츠 머신의 학습 알고리즘은 비감독 학습 방법인 기상-수면(wake-sleep) 알고리즘이다[6]. 기상-수면 알고리즘은 2단계 알고리즘으로 EM 알고리즘과 유사하게 가중치를 학습한다. 기상(wake) 단계에서는 생성 가중치만을 학습하는데 EM 알고리즘의 M 단계와 유사하고, 주어진 입력과 유사한 출력을 생성 가중치가 만들어 낼 수 있도록 한다. 수면(sleep) 단계에서는 반대로 “fantasy”라고 불리는 새로운 패턴을 생성하여 인식 가중치를 학습한다. EM 알고리즘의 E 단계와 유사하고, 은닉 노드가 보다 정확하게 패턴을 인식하도록 한다.

여러 가지 실험 결과를 보면[5, 6, 7], 숫자 패턴 인식 후 새로운 숫자 패턴 생성, 숨겨진 특정 패턴의 분석 등을 성공적으로 하는 것을 알 수 있다. 또한 압축에 사용할 경우에도 비교적 우수한 효율을 보이는 것을 볼 때, 헬름홀츠 머신은 패턴 분석에 적절한 모델이고 밀도 예측 능력을 가지고 있다는 것을 알 수 있다. 또한 생성 모델이기 때문에 새로운 샘플을 쉽게 생성할 수 있기도 하다. 그러므로 헬름홀츠 머신은 탐색적 분포 학습 알고리즘에 적합한 모델이라고 할 수 있다. 그리고 각 노드에서 자연스럽게 실수 벡터값들을 처리할 수 있기 때문에 실수 벡터 함수 문제에도 쉽게 적용할 수 있다는 장점이 있다.

3. 연속 함수 최적화 알고리즘

본 논문에서 제안하는 알고리즘에 대한 개략적인 설명이 표 1에 나타나 있다. 초기 데이터를 생성하고, 헬름홀츠 머신을 초기화한 후 헬름홀츠 머신을 2절에서 설명한 기상-수면 알고리즘을 이용해서 학습을 시킨다. 학습이 종료되면 새로운 데이터 집합을 생성하고 다음 세대 학습에 사용될 데이터를 $(\mu + \lambda)$ 절단 선택 방법과 엘리트 전

표 1. Helmholtz machine(HM)을 이용한 Estimation of Distribution Algorithm

<p>EDA using HM</p> <p>$D_0 \leftarrow$ Generate N individuals (the initial population) randomly</p> <p>Repeat for $l = 1, 2, \dots$ until a stop criterion is met</p> <p>$D_{l-1}^s \leftarrow$ Select $S \leq N$ individuals from D_{l-1} according to a truncation selection</p> <p>$p_l(x) = p(x D_{l-1}^s) \leftarrow$ Estimate the probability distribution of individual using Helmholtz machine</p> <p>$D_l \leftarrow$ Sample N individuals (the new population) from $p_l(x)$ (in Helmholtz machine)</p>

략을 이용해서 선택한다. 그리고 선택된 데이터 집합을 이용해서 학습을 반복한다. 기상-수면 알고리즘을 이용해서 헬름홀츠 머신을 학습을 할 때 가중치 변경은 지역 델타 규칙(local delta rule)을 사용하였다[8], 구체적인 식은 식 (1) ~ (6)에 설명되어 있다.

$$\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{v} \quad (1)$$

$$(\tau_j^2)' = \alpha \tau_j^2 \quad (2)$$

$$\mathbf{x} = \mathbf{G}\mathbf{y} + \boldsymbol{\varepsilon} \quad (3)$$

$$(\sigma^2)' = \alpha \sigma^2 \quad (4)$$

$$\mathbf{G}' = \mathbf{G} + \eta(\mathbf{x}^{(e)} - \mathbf{G}\mathbf{y}^{(e)})\mathbf{y}^{(e)} \quad (5)$$

$$\mathbf{R}' = \mathbf{R} + \eta(\mathbf{y}^{(f)} - \mathbf{G}\mathbf{x}^{(f)})\mathbf{x}^{(f)} \quad (6)$$

식 (1)은 은닉 노드(\mathbf{y})의 값을 결정하는 식인데, 즉 데이터(\mathbf{x})에 내재되어 있는 패턴을 파악하는 식이다. \mathbf{R} 은 인식 가중치 벡터이고, \mathbf{v} 는 노이즈로 $N(0, \tau^2)$ 의 분포를 가진다. τ 의 초기값은 1이고, 식 (2)에 의해서 값이 변경된다. 식 (3)은 새로운 데이터를 생성하는 식이다. \mathbf{G} 는 생성 가중치 벡터이고, $\boldsymbol{\varepsilon}$ 은 역시 노이즈로 $N(0, \sigma^2)$ 의 분포를 가진다. σ 은 역시 초기값은 1이고, 식(4)에 의해서 결정된다. 식 (2)과 (4)의 α 는 노이즈 감쇄율이다. 식 (5)은 생성 가중치를 학습하는 식이고, 식 (6)은 인식 가중치를 학습하는 식이다. 식 (5)와 (6)에서 η 는 학습률이다.

실수 벡터를 학습하기 위해서 기존의 헬름홀츠 머신을 약간 변형하였다. 헬름홀츠 머신의 각 노드에서 시그모이드 함수를 사용하지 않고 노이즈만을 이용해서 확률 유닛으로 구현하였다. 노이즈 때문에 각 노드의 값이 주어진 문제의 변수범위를 초과하지 않도록 조절하였다.

4. 실험 및 결과

테스트 함수로 구형(Sphere) 함수와 Ackley 함수를 사용하였다. 각 함수의 정의는 다음과 같다.

$$f_{sphere}(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (7)$$

$$f_{Ackley}(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \quad (8)$$

두 함수 모두 x 의 범위는 $-20 \leq x \leq 20$ 이고, $n = 40$ 이다. 최적해도 역시 두 경우 모두 $\vec{x} = (1, 1, \dots, 1)$ 일 때, $f(x) = 0$ 이다.

제한한 알고리즘의 우수성을 증명하기 위해서 기본적인 유전자 알고리즘과 비교하였다. 유전자 알고리즘은 실수 표현 방법을 사용하고, 기본적인 일점 교차 연산과 균일 분포(uniform distribution)를 이용한 돌연변이 연산, 적합도 비례 선택 방법을 사용하였다[9]. 실험시 사용한 각 변수들은 표 2와 표 3에 나타나 있고, 각 함수에 대해서 모두 10번씩 실험을 하였다.

헬름홀츠 머신을 이용한 탐색적 최적화 알고리즘의 구형 함수에 대한실험 결과는 그림 2에 설명되어 있고, Ackley 함수의 결과는 그

표 2. 제안한 최적화 방법 변수

parameters	값
학습 데이터 집합 크기(μ)	103
생성 데이터 집합 크기(λ)	105
HM 학습 반복 횟수	103
최대 세대 수	10
은닉 노드 수	1
입력 노드 수	40
노이즈 감쇄율(α)	0.999
학습률	0.001

표 3. 유전자 알고리즘 변수

parameters	값
최대 세대 수	10^9
개체군 크기	10^3
교차 연산자 비율	0.5
돌연변이 연산자 비율	0.1
염색체 크기	40

림 3에 나타나 있다. 실수 표현 방법을 사용한 유전자 알고리즘의 결과는 그림 4와 그림 5에 나타나 있다. 그림에서 볼 수 있듯이 제안한 알고리즘이 유전자 알고리즘보다 아주 좋은 결과를 보여주는 것을 알 수 있다. 10번의 실험 중 발견한 최적해는 구형 함수의 경우 탐색점 최적화 알고리즘은 6.12×10^{-9} 이었고, 유전자 알고리즘은 0.236이었다. Ackley 함수는 탐색점 최적화 알고리즘의 경우에는 1.67×10^{-6} 이었으며, 유전자 알고리즘은 0.578이었다.

특히 구형 모델처럼 단순한 경우는 물론 Ackley 함수처럼 변수들 간의 의존성이 있는 경우에도 만족할만한 결과를 얻는 것을 알 수 있다. 또한 7-8세대만에 어느 정도 수렴을 하는 것으로 보아 모델 학습에 소요되는 시간을 고려하더라도 최적해를 얻기까지 그리 많은 시간을 필요로 하지 않는다는 것도 발견할 수 있었다.

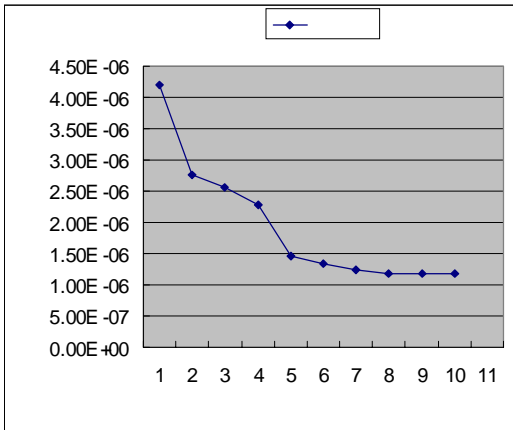


그림 2. 헬름홀츠 머신을 사용한 탐색점 분포 학습 알고리즘의 구형 함수 결과

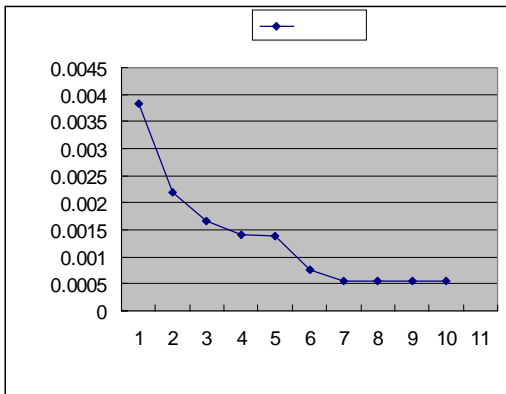


그림 3. 탐색점 분포 학습 알고리즘의 Ackley 함수 결과

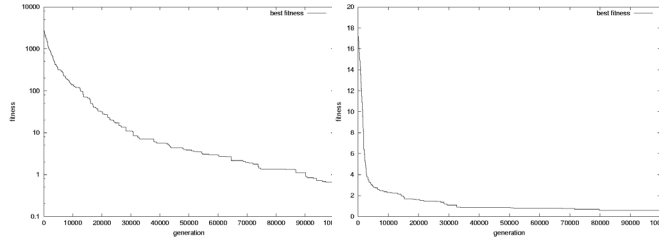


그림 4. 실수 표현형 유전자 알고리즘의 구형 함수 결과

그림 5. 유전자 알고리즘의 Ackley 함수 결과

5. 결론 및 향후 과제

본 논문에서는 기존의 헬름홀츠 머신을 이용한 탐색점 분포 학습 알고리즘을 확장하여 연속 함수의 최적화 문제를 해결해 보았다. 실험 결과에서 보여지는 것처럼 구형 함수 및 Ackley 함수에서 실수 표현형을 사용한 유전자 알고리즘보다 훨씬 우수한 결과를 얻을 수 있었다.

앞으로는 제안하는 알고리즘의 우수성을 보다 확실하게 검증하기 위해서 다양한 테스트 함수들에 대해서 실험을 해 보고, 진화 전략 (Evolutionary Strategy) 및 연속 함수의 최적화를 수행할 수 있는 기존의 탐색점 분포 학습 알고리즘의 결과와 비교해 보고자 한다.

감사의 글

본 연구는 과학기술부 뇌연구개발사업(BR-2-1-G-06)과 BK21 프로그램에 의하여 일부 지원되었음.

참고문헌

- [1] P. Larrañaga, R. Etxeberria, J. A. Lozano, B. Sierra, I. Inza, and J. M. Peña, "A review of the cooperation between evolutionary computation and probabilistic graphical models", *Proc. of the Second Symposium on Artificial Intelligence*, pp. 314-324, *Adaptive Systems*, 1999.
- [2] M. Pelikan and F. Lobo, "A Survey of Optimization by Building and Using Probabilistic Models", University of California, Urban Champaign Report, *IlligAL Report No. 99018*, 1999.
- [3] P. Larrañaga, R. Etxeberria, J. A. Lozano, B. Sierra, I. Inza, and J. M. Peña, "Optimization in continuous domains by learning and simulation of Gaussian networks", *Proc. of the Workshop in Optimization by Building and using Probabilistic Models, in GECCO-2000*, pp. 201-204, Morgan Kaufmann, 2000.
- [4] B.-T. Zhang and S.-Y. Shin, "Bayesian Evolutionary Optimization using Helmholtz Machines", *Lecture Notes in Computer Science (PPSN-2000)*, pp. 827-836, Springer, 2000.
- [5] P. Dayan, G. E. Hinton, R. Neal, and R. S. "Zemel, Helmholtz Machine", *Neural Computation*, **7**: 1022-1037, 1995.
- [6] G. E. Hinton, P. Dayan, B. J. Frey, R. M. Neal, "The "Wake-Sleep" Algorithm for Unsupervised Neural Network", *Science* **268**: 1158-1161, 1995.
- [7] B. J. Frey, G. E. Hinton, and P. Dayan, "Does the wake-sleep algorithm learn good density estimators?" *Advances in Neural Information Processing Systems*, Volume 8, pp. 661-667, MIT Press, 1996.
- [8] R. M. Neal and P. Dayan, "Factor Analysis Using Delta-Rule Wake-Sleep Learning", *Neural Computation*, **9**: 1781-1803, 1997.
- [9] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Inc., 1997. (pp. 60)