

순차적 베이지안 진화 연산을 이용한 시계열 예측

조동연^U 장병탁
서울대학교 컴퓨터공학부
{dycho, btzhang}@scai.snu.ac.kr

Sequential Bayesian Evolutionary Computations for Time Series Prediction

Dong-Yeon Cho^U Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

본 논문에서는 시간이 흐름에 따라 관측되는 시계열 데이터에 대한 예측을 위한 순차적 베이지안 진화 연산 기법을 제안한다. 이 방법에서는 이전 세대의 모델을 바탕으로 예측을 수행하고 새로운 데이터가 주어지면 현재의 예측 모델을 평가하여 더 좋은 모델을 생성하도록 한다. 제안된 방법을 시계열 데이터에 적용한 결과 기존의 방법보다 데이터에 적합한 모델을 학습하고 성공적인 예측을 수행함을 확인하였다.

1. 서론

학습은 제한된 수의 관찰된 데이터를 사용하여 입력과 출력 사이의 관계를 가장 잘 설명하는 모델을 찾는 과정이다. 여기서 모델이란 입력이 주어지면 그 입력을 처리하여 출력을 내보내는 시스템으로 생각할 수 있다. 어떤 모델을 결정하기 위해서는 모델의 구조와 그 구조에 따른 매개변수(parameter)들을 정해야 한다.

모델의 알맞은 구조와 매개변수를 동시에 탐색하는 문제에 유용하게 사용될 수 있는 방법이 베이지안 추론(Bayesian inference)이다. 이 방법에서는 모델에 관한 사전 지식과 주어진 데이터를 사용하여 얻은 지식을 확률 분포로 표현하고 베이스 정리(Bayes theorem)에 의하여 이 두 지식을 통합함으로써 주어진 데이터에 대한 모델의 확률 분포를 추정할 수 있게 된다. 그러나 원하는 확률 분포를 구하기 위해 다차원 함수의 적분을 계산해야 하는데 대부분의 경우에 이러한 적분을 구하는 것은 불가능하기 때문에 실제 응용에는 많은 제약이 있었다. 최근들어 베이지안 추론에서의 복잡한 적분 계산을 대체하는 방법으로 마르코프 사슬 몬테 칼로(Markov chain Monte Carlo, MCMC) 방법이 널리 사용되고 있으며[1], 여러 분야에서 좋은 성능을 보여 주고 있다[2, 3].

공학적 측면에서 모델의 알맞은 구조와 매개변수를 동시에 찾는 방법으로 널리 사용되어 온 것이 진화 연산(evolutionary computation)이다. 진화 연산은 자연 선택(natural selection)과 유전학에 기반한 방법으로 매우 큰 탐색 공간을 갖는 문제에

대한 해를 찾는 데 유용하게 사용된다[4]. 이러한 진화 연산이 베이지안 추론을 통하여 적합한 모델을 탐색하는 과정을 수행하는 알고리즘이 제시되었다[5].

이러한 방법들은 모두 데이터가 일괄적으로 주어졌을 때 데이터에 적합한 모델을 찾게 된다. 그러나 우리가 접하는 많은 데이터들은 시간에 따라 순차적으로 얻어진 것이다[6]. 예를 들면 시간에 따라 측정된 시스템의 출력 값이나 주가나 환율과 같은 경제 지표, 움직이는 물체의 위치를 관찰한 결과 등이 있다. 이러한 상황에서 기존의 방법을 사용하여 좋은 해를 찾는 것은 쉽지 않다. 본 논문에서는 신경 트리를 이용하여 시계열 예측 문제를 해결하기 위한 순차적 베이지안 진화 연산 기법을 제안한다.

논문의 구성은 다음과 같다. 2절에서는 시계열 예측을 위해 사용되는 모델인 신경 트리의 구조를 살펴보고 3절에서는 시계열 예측을 위해 순차적 베이지안 진화 알고리즘을 이용한 신경 트리의 학습 방법을 설명한다. 4절에서는 제안된 방법을 통해 수행한 시계열 예측의 성능을 평가하고 5절에서 결론을 맺는다.

2. 신경 트리의 구조

신경 트리는 다층 신경망을 트리 구조로 표현한 것으로 비단말 노드(nonterminal node)와 단말 노드(terminal node) 그리고 각 노드간의 연결 가중치로 구성된다[7]. 비단말 노드는 신경 유닛(neural unit)을 나타내고 각 신경 유닛의 형태는 기본 합

수 집합(function set) F 에서 선택된다. 단말 노드는 단말 집합 (terminal node) $T = \{x_1, x_2, \dots, x_n\}$ 에서 선택되는데 여기서 x_i 는 입력 \mathbf{x} 의 i 번째 성분이다. 링크(link) (j, i) 는 노드 j 로부터 노드 i 로의 연결을 나타내며, 각 링크에는 연결 가중치 w_{ji} 가 존재한다. 이러한 구조의 신경 트리에서 루트(root) 노드가 출력 노드가 되고 각 단말 노드들이 입력 노드가 된다.

비단말 노드는 최대 b_{max} 개의 자식 노드로부터 입력을 받아 하나의 출력을 내보내게 되는데, 신경 유닛의 형태에 따라 그 입력을 받아들이는 방법이 다르다. 가장 대표적인 유닛은 시그마(Σ) 유닛과 파이(Π) 유닛으로 다음과 같이 모든 자식 노드로부터의 입력의 가중 합과 가중 곱을 구한다.

$$net_i = \sum_j w_{ji}y_j, \quad net_i = \prod_j w_{ji}y_j \quad (1)$$

여기서 y_j 는 자식 노드 j 로부터의 입력이다. 신경 유닛의 출력은 식 (2)의 시그모이드(sigmoid) 함수에 의하여 계산된다.

$$y_i = f(net_i) = \frac{1}{1 + e^{-net_i}} \quad (2)$$

여기서 net_i 는 식 (1)에 의하여 계산된 값이다.

3. 순차적 베이지안 진화 연산

3.1 확률 분포의 정의

어떤 신경 트리 A 가 주어졌을 때 그 사후 확률은 다음과 같이 정의된다.

$$P(A|D) \propto P(D|A)P(A) = P(D|\mathbf{w}, k)P(\mathbf{w}, k) = P(D|\mathbf{w}, k)P(\mathbf{w}|k)P(k) \quad (3)$$

여기서 k 는 신경 트리의 노드의 개수이고 \mathbf{w} 는 가중치의 벡터이다. 훈련 데이터가 다음과 같이 주어지고,

$$D = \{(\mathbf{x}_c, y_c)\}_{c=1}^T \quad (4)$$

신경 트리 A 에 의하여 구현되는 함수가 f_A 라 하면,

$$y_c = f_A(\mathbf{x}_c) + \epsilon \quad (5)$$

가 성립한다. 여기서 ϵ 은 주어진 신경 트리의 에러로 평균이 0이고 표준편차가 σ 인 정규 분포를 따른다고 가정한다. 이 경우 각 데이터가 서로 독립이라고 가정하고, 가중치 벡터를 이루는 각 성분이 평균이 0이고 표준편차가 1인 정규 분포를 따르며 서로 독립이라하며, 또한 노드 개수는 평균이 λ 인 포아송(Poisson) 분포를 따른다고 가정하면, 어떤 신경 트리 A 의 사후 확률은 다음과 같이 정의된다.

$$P(A|D) \propto P(D|\mathbf{w}, k)P(\mathbf{w}|k)P(k) = \left\{ \frac{1}{\sqrt{2\pi}\sigma} \right\}^T \exp \left\{ -\frac{\sum_{c=1}^T (y_c - f(\mathbf{w}, k)(\mathbf{x}_c))^2}{2\sigma^2} \right\} \times \left\{ \frac{1}{\sqrt{2\pi}} \right\}^{k-1} \exp \left\{ -\frac{\sum_{j=1}^{k-1} w_j^2}{2} \right\} \frac{(\lambda-3)^{k-3} \exp(-\lambda+3)}{(k-3)!} \quad (6)$$

단, 단말 노드만으로 이루어진 신경 트리는 고려하지 않는다. ($k = 3, 4, 5, \dots$)

3.2 순차적 베이지안 진화 연산에 의한 시계열 예측

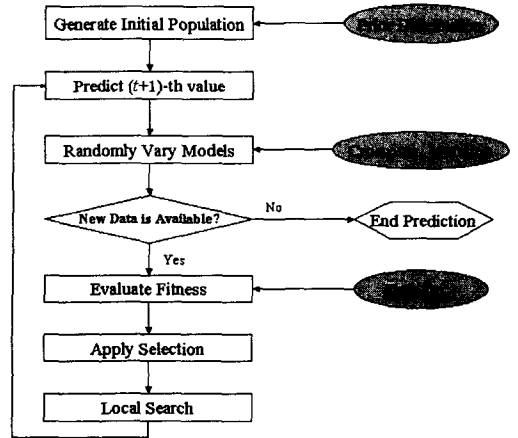


그림 1. 순차적 베이지안 진화 연산의 흐름도

주어진 데이터에 적합한 신경 트리의 구조와 매개변수 즉 연결 가중치를 찾기 위해 다음과 같이 신경 트리 A_t 로 구성된 개체군 $A(t)$ 를 유지한다.

$$A(t) = \{A_1, A_2, \dots, A_M\} \quad (7)$$

여기서 M 은 개체군의 크기를 나타낸다. 초기 개체군 $A(0)$ 는 모델의 사전 확률에 따라 생성된다.

$(t+1)$ 번째의 값은 $A(t)$ 에 속하는 개체들의 출력 값을 평균하여 예측한다.

$$\hat{y}_{t+1} = \frac{1}{M} \sum_{i=1}^M f_{A_i}(\mathbf{x}_{t+1}), \quad A_i \in A(t) \quad (8)$$

새로운 모델을 생성하기 위하여, 부모에서 임의로 선택된 부트리를 서로 교환하는 교차 연산과 역시 임의로 선택된 부트리에 대하여 각 신경 유닛의 형태를 바꾸거나 단말 노드에서 입력의 성분을 다른 것으로 교체하는 돌연변이 연산을 사용된다.

$(t+1)$ 번째의 실제 값이 주어지면 최근 k 개의 데이터와 새로 추가된 데이터를 사용하여 생성된 모델의 성능을 평가한다. 즉 새로운 개체군 내의 모든 개체에 대하여 제곱 오차를 다음과 같이 계산하고,

$$E^2(t+1) = \sum_{c=t-k+1}^{t+1} (y_c - f_{A_i}(\mathbf{x}_c))^2, \quad A_i \in A(t+1) \quad (9)$$

이 값을 이용하여 식 (6)의 사후 확률을 계산한다.

다음 값을 예측하기 위해 사용될 개체들을 적합도에 비례하여 선택한다. 여기서 사후 확률이 적합도로 사용되며 비례 선택을 위하여 아래와 같이 정규화를 한다. 즉, 각 개체는 식 (10)의 확률로 선택된다. (단 $D_{t+1} = \{y_{t-k+1}, y_{t-k+2}, \dots, y_t, y_{t+1}\}$)

$$P_{t+1}(A_i|D_{t+1}) = \frac{P_{t+1}(A_i|D_{t+1})}{\sum_{j=1}^M P_{t+1}(A_j|D_{t+1})}, \quad A_i \in A(t+1) \quad (10)$$

선택된 신경 트리의 가중치는 언덕 오르기(hill-climbing)에 의하여 조정된다. 가중치의 변경은 가중치 벡터 \mathbf{w} 의 모든 성분에 대하여 임의의 순서로 이루어지는데 이 때 사용되는 변경식은 다음과 같다.

$$w'_j = w_j + N(0, 1) \quad j=1, 2, \dots, k_i - 1 \quad (11)$$

여기서 k_i 는 개체 A_i 의 노드의 개수이며 $N(0,1)$ 은 평균이 0이고 표준편차가 1인 정규 분포를 따르는 값이다. 이렇게 하나의 가중치가 변경될 때마다 역시 식 (9)의 제곱 오차를 계산하고 오차가 줄어든 경우에만 새로운 가중치를 수용한다. 그림 1은 순차적 베이저안 진화 연산의 일반적인 흐름을 나타내고 있다.

4. 실험 및 결과

이 데이터는 1700년부터 1999년까지의 연평균 태양 흑점 (sunspot)의 수를 기록한 것이다. b_{max} 는 입력의 개수와 같은 4로 하였으며, 에러의 표준 편차 σ 는 0.05, 트리의 평균 노드 수 λ 는 40으로 설정하였다. 진화 연산을 수행할 때 교차 연산의 확률은 2/3, 돌연변이 연산의 확률은 1/3이었고, 사용된 개체군의 크기 M 은 1,000이었다. 그림 2는 제안된 방법에 의하여 태양 흑점 수를 예측한 결과를 보여 주고 있다. 초기의 예측 결과를 제외하면 실제 값과 유사한 예측 값을 생성해 낼 수 있다.

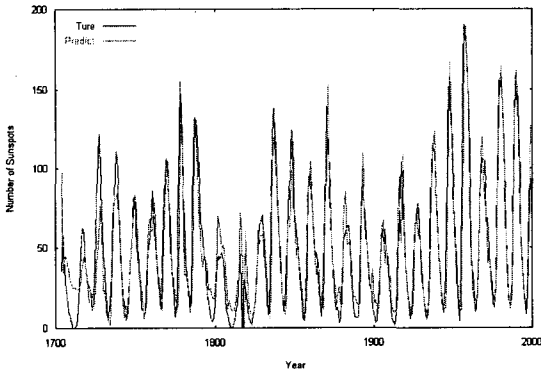


그림 2. 태양 흑점 수의 예측 결과 ($k=50$)

표 1은 본 논문에서 제안된 순차적인 방법과 일반적인 진화 연산에서 사용하는 일괄 처리 방식의 예측 오차(NMSE)를 비교하고 있다. 여기서 일반적인 진화 연산도 역시 베이저안 진화 연산으로 구현되었으며 다른 모든 실험 설정도 동일하다. 단, 일괄 처리 방식의 진화 연산에서는 처음에 주어진 k 개의 데이터에 대하여 개체들을 적합도를 평가하고 일정 세대가 지난 후 가장 좋은 모델을 선택하여 새로운 데이터에 대한 예측을 수행한다.

실험 결과, 순차적인 진화 연산에 의한 예측 결과가 일괄 처리 방식의 예측 결과보다 오차가 적으며 더 안정된 성능을 얻을 수 있었다. 일괄 처리 방식의 진화 연산에서는 주어진 데이터에 의하여 얻어진 결과를 계속해서 사용하기 때문에 나중에 들어오는 데이터에 대해서는 적용하지 못하여 오차가 커지게 된다. 반면에, 순차적인 베이저안 진화 연산은 새로운 데이터를 계속해서 개체의 적합도를 평가할 때 사용하므로 시간이 흐름에 따라 적합한 모델을 생성할 수 있게 된다. 이러한 효과는 사용되는 데이터의 개수가 적은 경우 더욱 분명히 나타난다.

표 1. 순차적 방법과 일괄 처리 방법의 비교 (10번 실험)

방법	$k = 30$		
	평균±표준편차	최소	최대
Sequential	0.21496±0.02697	0.18932	0.28179
Batch	0.23775±0.03812	0.18719	0.31566

방법	$k = 50$		
	평균±표준편차	최소	최대
Sequential	0.20747±0.02139	0.16730	0.25148
Batch	0.21525±0.02857	0.16867	0.28179

5. 결론

기존의 대다수 학습 방법은 주어진 데이터만을 사용하여 적합한 모델을 찾기 위해 노력하였다. 그러나 실제 세계에서 얻어지는 많은 데이터들은 시간의 흐름에 따라 순차적으로 관찰되며 새롭게 얻어진 데이터는 이전의 데이터와 다른 특성을 갖게 되는 경우가 있다. 본 논문에서는 이러한 환경에서 적합한 모델을 찾기 위한 순차적 베이저안 진화 연산을 제안하였고, 시계열 예측 문제에 적용하여 기존의 방법보다 좋은 성능을 얻었다.

제안된 방법은 환경이 변하는 상황에서 능동적으로 대처해야 하는 제어 시스템이나 이동하는 물체의 위치를 파악하여 추적하는 감지 장치의 학습에도 응용될 수 있을 것이다.

감사의 글

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

참고문헌

- [1] Gilks, W.R., Richardson, S., and Spiegelhalter, D.J., *Markov Chain Monte Carlo in Practice*, Chapman & Hall, 1996.
- [2] Neal, R.M., *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics 118, Springer, 1996.
- [3] Andrieu, C, de Freitas, N, and Doucet, A., "Robust full bayesian methods for neural networks", *Advances in Neural Information Processing Systems*, Vol.12, pp. 379-385, MIT Press, 2000.
- [4] Fogel, D.B., "What is evolutionary computation?", *IEEE Spectrum*, Vol 37, No. 2, pp. 26-32, 2000.
- [5] Zhang, B.-T., "A Bayesian framework for evolutionary computation," *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol 1, pp. 722-728, 1999.
- [6] 조신섭, 손영숙, *시계열 분석*, 율곡출판사, 1999.
- [7] Zhang, B.-T., Ohm, P., and Mühlenbein, H., *Evolutionary Neural Trees for Modeling and Predicting Complex Systems*, *Engineering Applications of Artificial Intelligence*, Vol. 10, No. 5, pp. 473-483, 1997.