다 입력 퍼지 변수를 위한 자기 학습 퍼지 알고리즘

A Self Learning Fuzzy Algorithm for Multi-Input Fuzzy Variables

김광용* , 윤호섭, 소정, 민병우
한국전자통신연구원 컴퓨터소프트웨어연구소 영상처리연구부

Kwang-Yong Kim, Ho-Sub Yoon, Jung Soh and Byung-Woo Min
Image Processing Department, ETRI - Computer S/W Technology Lab.

요 약

입•출력 데이터 쌍만을 이용하여 규칙 및 소속 함수를 자동적으로 결정하는 자기 학습 퍼지 알고리즘 중에서, 가장 이해하기 용이하고 퍼지 규칙 및 소속 함수 생성이 빠른 방법으로 기울기 강하를 이용한 방법들이 있다. 기울기 강하를 이용한 방법중에서 가장 대표적인 Araki가 제안한 방법은 퍼지 조건부가 퍼지 집합 형태이고 결론부는 단일값으로 구성된 알고리즘으로써 입력 퍼지 공간을 세분화하면서 시스템을 규명해나가는 간단하면서도 효율적인 알고리즘이다. 그러나 이 방법은 퍼지 입력 변수가 증가하면 퍼지 공간이 세분화 되면서 소속 함수 및 규칙 생성 개수가 급격히 제곱배로 증가하는 문제점을 가지고 있다. 따라서, 본 논문에서는 퍼지 입력 변수가 증가함에 따라 급격히 퍼지 규칙 및 소속 함수의 수가 증가하는 Araki 알고리즘의 문제점을 분석하여 소속 함수 및 규칙 수의 급격한 증가를 억제하고 Araki 방법에 비해 학습 속도가 현저히 향상된 새로운 방안을 제안한다. 연구 결과, Araki 방법이 입력 변수의 개수가 증가 할수록 규칙 수가 기하 급수적으로 많이 필요하였던 것에 비해 제안한 방법은 훨씬 적은 규칙 수로 우수한 성능을 얻을 수 있었다.

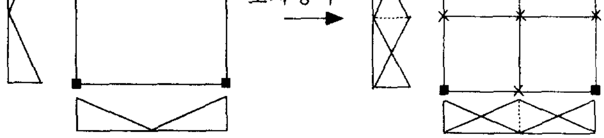
1. 서론

퍼지 논리는 비선형 특성을 가지는 시스템에 적용할 때, 계산이 복잡하지 않고 언어적인 규칙으로 수정이 용이하다는 장점 때문에 효과적으로 이용되어 왔다. 그러나 퍼지 시스템을 설계할 때 문제가 되어 왔던 것은 초기 설계시, 여러 차례의 시행 착오가 필요했다는 것이다. 이와 같은 문제점을 해결하기 위해서 인간이 학습을 통해 새로운 지식과 경험을 얻어 내는 것과 같이 스스로 퍼지 규칙을 생성하고 퍼지 소속 함수를 동조할 수 있는 자기생성 퍼지 시스템 개발이 필요하게 되었다. 이와 관련한 자기생성 퍼지 시스템 중에서 비교적 알고리즘이 이해하기 쉽고 학습 속도가 빠른 응답 특성의 오차를 최소화 하는 방향으로 기울기 강하(gradient descent)를 통해 퍼지 시스템을 규명해 내는 방법들이 등장하였다.[1-8]. 그 중에서 Araki[1]가 개발한 자기생성 퍼지 알고리즘은 퍼지 입력 변수가 증가하면 퍼지 공간이 세분화되면서 소속 함수 및 퍼지 규칙 수가 제곱배로 생성되는 문제점이 있었다. 이와 같은 문제점을 해결하기 위해 퍼지 규칙 수 및 소속 함수의 수를 현저히 최소화할 수 있는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구 및 제안한 방법을 알아보고 제안한 퍼지 알고리즘의 처리과정을 그림으로 도시하여 특성 및 장점을 살펴본다. 그리고 3장에서는 제안한 알고리즘을 이용하여 함수 근사화에 대해 Araki방법과 결과를 비교하고 결론을 맺는다.

2. 관련연구 및 제안한 방법

<그림 1>은 기울기 강하에 바탕을 둔 대표적인 자기생성 퍼지 알고리즘인 Araki[1]가 제시한 방법이 있다. 이 방법은 2차원 변수일 경우, 퍼지 입력 공간을 분할할 때 소속 함수의 좌 또는 우측 폭을 인접한 두 소속 함수의 중앙값 사이의 간격과 동일하게 분할하는 방법이다.



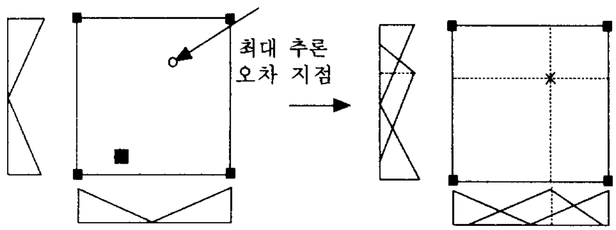
<그림 1> Araki 방법에 의한 퍼지 입력 공간의 분할

<그림1>에서 ' ' 은 기존의 퍼지 규칙을 표시한 것이고 'x'는 새로 생성된 퍼지 규칙을 의미한다.

이때 세분화 하는 방법은 퍼지 입력 공간에서 입출력 학습 데이터를 이용하여 평균적으로 가장 큰 추론 오차를 발생하는 영역을 찾는다. 그리고 나서 이 영역을 균등하게 2등분 한다.

그런데 <그림 1>에서 보는 바와 같이 Araki방법은 각 입력 변수마다 소속 함수 2개에서 시작하였다면, 다음 생성 단계에서 각각 2등분 되면서 3 X 3 - 2 X 2 = 5개의 새로운 퍼지 규칙 수가 급격히 생성되는 문제점을 가지고 있다. 이와 같은 퍼지 규칙 수의 급격한 증가는 입력 변수가 많을수록 문제가 생긴다.

따라서 다음과 같은 방법으로 퍼지 규칙 수 및 소속 함수의 급격한 증가를 해결하는 알고리즘을 제안한다.



<그림 2> Kim방법에 의한 퍼지 입력 공간 분할

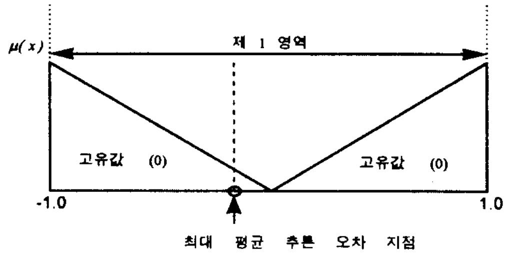
여기서, ' '는 기존의 퍼지 규칙을 보인 것이고 'x'는 새로 생성된 퍼지 규칙을 표시한 것이다.

<그림 2>는 제안한 방법을 도시한 것으로 새로운 소속 함수를 생성할 때, 퍼지 규칙 수가 오직 1개씩 증가한다는 특징을 가지고 있다. 특히, 이 방법은 퍼지 입력 공간을 계속 세분화 해 나가는 것이 아니라 최대 푸론 오차를 발생시킨 해당 영역이 어느 곳에 위치해 있느냐에 따라 삽입 생성하면서 분할하거나 기존의 소속 함수의 좌 또는 우측 폭만을 조금씩 증가 시켜 나가는 방법이다. <그림 2>에서 보는 바와 같이 새로운 소속 함수는 인접한 소속 함수와 불균등하게 중첩되어 생성 되어진다.

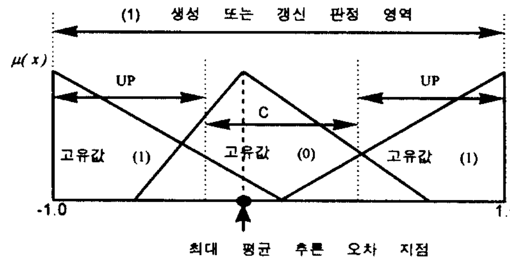
이와 같이 제안한 방법은 기존 Araki방법이 가진 퍼지 규칙 수의 기하 급수적 증가라는 문제점을 해결함으로써 다 입력 변수 시스템의 경우, 적은 규칙 수로 우수한 성능을 얻어낼 수 있다.

제안한 알고리즘의 소속 함수 생성 및 갱신하는 과정은 다음 <그림 3>에서부터 <그림 10>을 통해 알 수 있다. 여기서 갱신(update)이란 새로운 소속 함수를 인접한 소속 함수 사이에 무조건 삽입(insertion)하는 것이 아니라 인접한 좌 또는 우측 소속 함수의 폭만을 최대 평균 추론 오차 영역 내의 추론 지점의 편중 정도에 따라 서서히 늘려나가는 것을 의미한다.

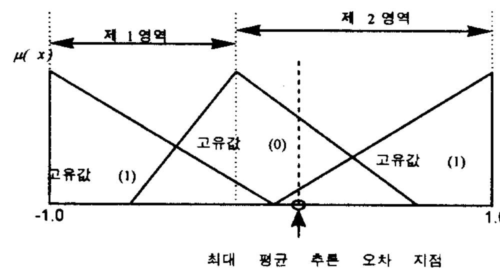
간단한 예로서 <그림 3>부터 <그림 10>은 1개의 입력 변수에 대해 최초 2개의 퍼지 소속 함수가 있을때의 처리 과정을 순서대로 나열한 것이다.



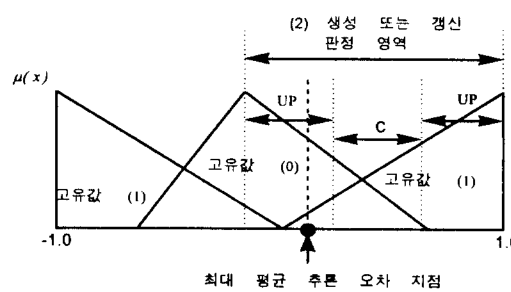
<그림 3> 최대 평균 추론 오차 지점 발생



<그림 4> 새로운 퍼지 소속 함수의 삽입

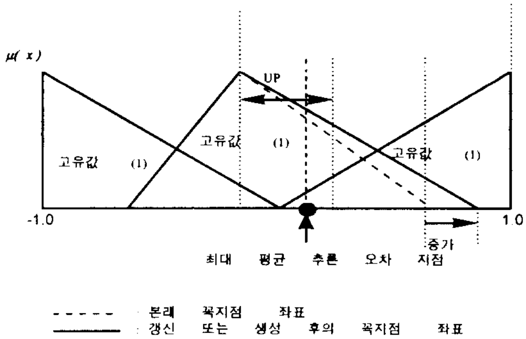


<그림 5> 두 퍼지 영역에서의 다음 추론 오차지점 발생

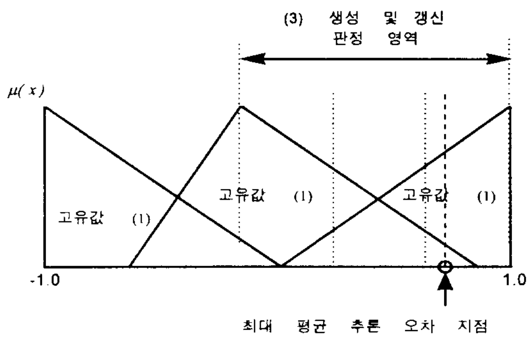


<그림 6> 소속 함수의 삽입 및 갱신 판정

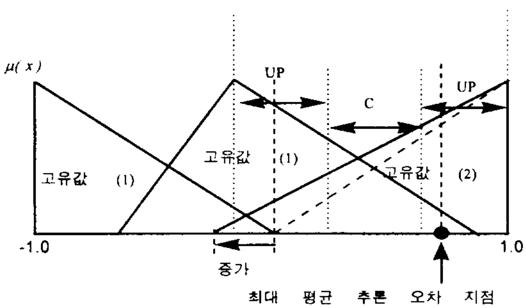
■ 그림에서 'UP'은 갱신 영역을 의미하고, 'C'는 생성 영역을 의미한다.



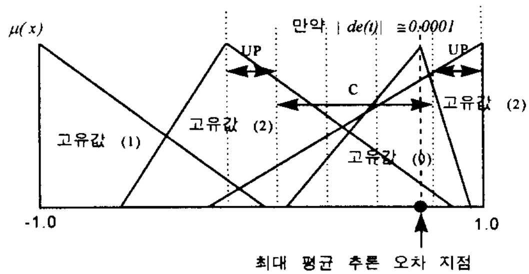
<그림 7> 기존 소속 함수의 갱신



<그림 8> 다음 최대 추론 오차 지점 발생



<그림 9> 기존의 소속 함수의 갱신



<그림 10> 한 퍼지 영역을 미세하게 판정

<그림 10>은 학습을 진행하다가 오차 미분값이 0.0001이하로 매우 느리게 오차 변화폭이 감속하게 될 경우이다.

이와 같이 오차 미분량이 극소한 경우, 해당 추론 오차 영역을 더욱 세밀하게 처리하기 위해 3등분에서 5등분으로 등분하여 갱신 및 생성 영역 판단 구역으로 정한다. 따라서, 등분의 수가 많아지면 n등분한 영역중 최초 1등분과 마지막 n번째 등분 영역만이 소속 함수 갱신에 이용하고 나머지 구역은 생성 영역의 확률을 높임으로써 세밀하게 시스템을 규명할 수 있도록 한 것이다.

3. 실험결과 및 고찰

실험 대상은 임의 비선형 함수식에 대해 입력 변수가 1개일 경우에서 3개일 경우까지 함수 근사화(function approximation)실험을 하였다.

실험 대상으로 이용한 비선형 함수식은 다음과 같다.

- 함수 근사화

$$w = -x\ (-1<x\le 0),\quad w = x^2,\ (0<x<1) \tag{1}$$

$$w = 2^{-\frac{2}{3}}\cdot(1.27x+0.27)^2\cdot\{1-(1.27x+0.27)\}^{\frac{2}{3}},\ (-1<x<1) \tag{2}$$

$$w = \sqrt{1-x^2-y^2},\ (-1<x,y<1) \tag{3}$$

$$w = \frac{\sqrt{x^2+y^2+z^2}}{3},\ (-1<x,y,z<1) \tag{4}$$

$$w = (\sin(\pi x)+\cos(\pi y)+\sin^{-1}(\pi z)),\ (-1<x,y,z<1) \tag{5}$$

여기서, $-1<x, y, z<1$ 이다.

이 식에서 (1),(2)식은 1차원 변수에 대한 실험이고 (3)식은 2차원, 그리고 다차원 식에서의 실용성을 입증하기 위해 (4),(5)식은 3차원 변수에 대한 실험식이다.

표 1. Araki방법과 제안한 방법의 규칙 수 비교

수식의 종류	초기 규칙수	총 규칙수
(1) Araki방법	2	14
제안한 방법	2	3
(2) Araki방법	2	5
제안한 방법	2	4
(3) Araki방법	4	1089
제안한 방법	4	9
(4) Araki방법	8	15625
제안한 방법	8	28
(5) Araki방법	8	19683
제안한 방법	8	36

표2. Araki방법과 제안한 방법의 소속 함수 비교

수식의 종류	초기 소속 함수의 수	총 소속 함수의 수
(1) Araki방법	2	14
제안한 방법	2	3
(2) Araki방법	2	5
제안한 방법	2	4
(3) Araki방법	2 : 2	33 : 33
제안한 방법	2 : 2	7 : 5
(4) Araki방법	2 : 2 : 2	25 : 25 : 25
제안한 방법	2 : 2 : 2	12 : 16 : 13
(5) Araki방법	2 : 2 : 2	27 : 27 : 27
제안한 방법	2 : 2 : 2	17 : 21 : 16
(6) Araki방법	2 : 2	5 : 5
제안한 방법	2 : 2	3 : 3

결과적으로 표 1에서와 같이 기존의 Araki방법은 입력 차수가 증가할수록 퍼지 규칙수가 지수적으로 증가함을 볼 수 있다. 제안한 방법은 기존의 Araki방법에 비해 매우 적은 퍼지 규칙수를 생성한다. 또한 표2의 결과에서와 같이 초기에 입력 변수마다 소속 함수의 수를 2개로 동일하게 주고 시작했을 경우, 제안한 방법은 각 입력 변수에 대해 퍼지 소속 함수의 수가 서로 다르게 생성되나 기존 Araki방법은 각 입력 변수에 대해 소속 함수의 수가 동일하게 생성된다.

참고 문헌

[1] S.Araki et al., "A Self-Generating Method of Fuzzy Inference Rules", Fuzzy Engineering toward Human Friendly System (PartVIII), vol.2, pp.1047-1058,1991.

[2] K.Kim et al., "A New Gradient Descent Based Self Generating Fuzzy Algorithm Using the partition of Fuzzy Input Space," Proc. of International Conference on Neural Information Processing and Intelligent Information Systems, vol.1, pp.494-497, 1997.

[3] Y.Shi et al., "A learning Algorithm for Tuning Fuzzy Rules Based on the Gradient Descent Method," Proc. of IEEE International Conference on Fuzzy Systems, vol.1, pp.55 - 61, 1996.

[4] T.Takagi et al., "Fuzzy Identification of Systems and its Applications to Modeling and Control," IEEE Trans. on Systems,Man, and Cybernetics, vol.SMC-15, no.1, pp.116-132, 1985.

[5] H.Ichihashi, T.Watanabe, "Learning Control System by a Simplified Fuzzy Reasoning Model," IPMU'90, Paris-France, pp.417 - 419, 1990.

[6] H.Nomura et al., "A Self-Tuning Method of Fuzzy Control by Descent Method," Proc. of 4th IFSA Congress, Brussels, vol.Engineering, pp.155 -158, 1991.

[7] S.Romaniuk et al., "Learning Fuzzy Control Rules from Examples," Fuzzy Control Systems, CRC Press, pp.375 - 395, 1994.

[8] Y.Shi et al., "A learning Algorithm for Tuning Fuzzy Rules Based on the Gradient Descent Method," Proc. of IEEE International Conference on Fuzzy Systems, vol.1, pp.55 - 61, 1996.