

자기구성 클러스터링 기반 뉴로-퍼지 모델링

Neuro-Fuzzy Modeling based on Self-Organizing Clustering

김승석*, 유정웅*, 김용태**

Sung-Suk Kim*, Jeong-Woong Ryu*, Yong-Tae Kim**

*충북대학교 전기전자공학부, 컴퓨터 정보통신 연구소

**한경대학교 정보제어공학과, 전자기술종합연구소

*Chungbuk National University School of Electrical and Electronic Engineering

Research Institute for Computer and Information, Communication

**Department of Information & Control Engineering, Hankyong National University

요 약

본 논문에서는 클러스터링을 뉴로-퍼지 모델에 직접 적용하여 모델을 최적화하는 방법을 제안하였다. 기존의 오차미분기반 학습을 통한 뉴로-퍼지 모델의 최적화 과정과는 달리 제안된 방법은 클러스터링 학습과 연계하여 모델을 구성하며 자율적으로 클러스터의 수를 추정하며 동시에 최적화를 수행한다. 순차적인 학습 기법에서는 각각의 학습 기법을 따로 적용하여 모델링을 실시하였으나 제안된 기법에서는 하나의 클러스터링 학습으로 전체 모델의 학습을 실시하였다. 또한 제안된 방법에서는 클러스터링이 수렴하는 만큼 전체 모델의 연산량이 감소하여 학습과정에서 발생하는 연산량 문제를 개선하였다. 시뮬레이션을 통하여 기존의 연구 결과들과 비교하여 제안된 기법의 유용성을 보였다.

Abstract

In this paper, we propose a new neuro-fuzzy modeling using clustering-based learning method. In the proposed clustering method, number of clusters is automatically inferred and its parameters are optimized simultaneously. Also, a neuro-fuzzy model is learned based on clustering information at same time. In the previous modelling methods, clustering and model learning are performed independently and have no exchange of its informations. However, in the proposed method, overall neuro-fuzzy model is generated by using both clustering and model learning, and the information of modelling output is used to clustering of input. The proposed method improve the computational load of modeling using Subtractive clustering method. Simulation results show that the proposed method has an effectiveness compared with the previous methods.

Key words : Neuro-Fuzzy Model, Fuzzy Clustering, Maximum Likelihood Estimation, TSK Fuzzy Model

1. 서 론

뉴로-퍼지 시스템은 인간의 학습능력을 모사한 신경회로망과 의사 추론 능력을 모사한 퍼지 시스템을 융합하여 각각의 장점을 하나의 시스템으로 구현한다[1-3]. 뉴로-퍼지 시스템의 일반적인 구조는 퍼지 추론 시스템으로 구성되며 학습 방법 등을 신경회로망 기법을 이용한다[2]. 이 경우 모델의 학습은 구조 동정(Structure Identification)과 파라미터 동정(Parameter Identification)으로 나누어진다[1][2][5]. 모델의 형태를 결정하는 구조 동정에서 모델의 크기와 학습 형태가 결정이 되고 파라미터 동정을 통하여 이들 파라미터들이 최적화한다[16].

모델의 성능은 학습 모델의 구조와 학습방식에 영향을 받는다. 일반적인 뉴로-퍼지 시스템의 경우 입력을 퍼지화 하는 전제부와 이를 규칙을 통하여 추론하는 퍼지 추론부 및

이를 비퍼지화하는 결론부로 구성되어있다[1][2][5]. 입출력 데이터를 이용하여 모델을 구성할 때 모든 입력공간을 격자 분할(grid partition)하여 각각에 대하여 규칙을 부여하는 기법이 있다. 모든 입력공간을 모델에서 이용할 수 있는 장점이 있지만 입력공간 또는 전제부 소속함수의 수가 증가하는 경우 모델의 규칙 수가 지수함수적으로 증가하는 단점을 가진다[2][5]. 이러한 문제점을 개선하는 방법으로 클러스터링 기법을 이용하여 전제부 소속함수를 생성하고 모델의 규칙을 부여하는 방법이 활발히 연구되어지고 있다[2][4]. 이 경우 패턴의 수 또는 클러스터의 수를 결정하는 문제와 파라미터를 최적화 문제가 모델의 성능과 직접적으로 관련되어진다. 또한 이 방법은 클러스터링에 의하여 추정된 파라미터가 뉴로-퍼지 시스템의 초기치로 설정되어 다른 학습방법으로 최적화되며 클러스터링의 파라미터가 뉴로-퍼지 시스템에서 학습된 파라미터와 틀려져 상호간의 연관성을 설명하기 어려운 경우도 발생한다[4][7].

본 논문에서는 자율적으로 클러스터의 수를 추정하며 이를 이용하여 뉴로-퍼지 시스템을 구성하여 모델을 학습하는 기법을 제안한다. 클러스터링 과정에서 사전 임계치를 이용하여 클러스터의 수를 추정하고 동시에 이들 파라미터를 최

접수일자 : 2005년 8월 1일

완료일자 : 2005년 10월 24일

본 연구는 경기도 차세대 성장동력 기술개발사업의 지원을 받아 수행되었음. * 교신저자

적화한다. 알고리즘이 진행되는 동안 클러스터 수의 추정에서 발생하는 연산량 증가 문제를 개선된 Subtractive 클러스터링 기법을 이용하여 추정된 클러스터 수만큼 감소시킨다 [6-8]. 제안된 방법을 적용한 클러스터링 방법을 이용하여 기존의 오차미분기반 학습을 클러스터링 기반 학습 방법으로 확장한다. 클러스터링에서 추정된 정보를 직접 모델의 학습에 이용하고 결과를 다시 클러스터링 학습에 이용하는 기법을 통해 모델을 구성하고 학습하였다. 클러스터링 관점에서 보면 전형적인 비교사학습을 모델의 출력 정보를 이용하여 교사 학습으로 확장하여 클러스터링 뿐만 아니라 전체 모델의 학습을 실시하는 방식으로 클러스터링과 뉴로-퍼지 모델의 학습을 통합하였다.

본 논문의 구성은 다음과 같다. 2장에서 제안된 클러스터링의 학습 방식을 설명하였으며, 3장에서는 클러스터링을 이용하여 뉴로-퍼지 모델을 구성하고 학습하는 방법에 대해 설명하였다. 4장에서 제안된 기법을 시뮬레이션을 통하여 결과를 나타내었고, 마지막으로 결론에서 정리하였다.

2. 자기구성 클러스터링

제안된 방법은 클러스터 수의 추정과 파라미터 최적화 과정으로 나누어지며, 알고리즘 수행 연산량을 감소시켜 수렴 속도를 개선하는 부분으로 구성되어 있다.

2.1. 클러스터 수의 수렴 및 최적화

클러스터를 추정하기 위하여 먼저 초기 클러스터 중심과 데이터 간의 유사도를 측정하여야 한다. 유사도의 측정 방식에 따라 클러스터링의 성능이 달라질 수 있다. 제안된 기법에서는 데이터 $x_i, i=1, 2, \dots, N$ 에 대해 클러스터 $c_j, j=1, 2, \dots, m$ 을 추정하고자 할 때 공분산까지 고려하는 다음과 같은 유사도 측정함수를 이용하였다[9][10][18].

$$r_{ij} = \exp\left(-\frac{1}{2}(c_i - c_j)^T \Sigma_j^{-1}(c_i - c_j)\right) \quad (1)$$

여기서 유사도 r_{ij} 는 i 번째 데이터와 j 번째 클러스터 중심 사이의 유사도를 의미하고, Σ_j 는 j 번째 클러스터 중심의 공분산 행렬이다. 초기 클러스터 중심 c_j 는 데이터 x_i 의 값으로 초기화 된다. 현재 클러스터의 추정에 사전에 지정된 임계치 ζ 보다 적은 유사도를 가지는 데이터를 다음과 같이 영향을 제거한다.

$$r_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} > \zeta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

이때 ζ 의 값이 너무 커지는 경우, 바로 이웃한 데이터도 현재의 클러스터 추정에 영향을 미치지 못하는 경우가 발생하고 너무 작은 경우 ζ 의 필요성이 없어진다. 이러한 문제점을 해결하기 위하여 본 논문에서는 유사도 측정함수 식(1)을 다음과 같이 제한하였다.

$$r_{ij} = \exp\left(-\frac{1}{2}(c_i - c_j)^T (\zeta \cdot \Sigma_j)^{-1}(c_i - c_j)\right) \quad (3)$$

ζ 의 크기에 따라 유사도 측정 범위가 달라지는데 식(2)에서 ζ 를 너무 작게 설정하는 경우 ζ 의 필요성이 없어지게 되

며 너무 크게 설정하는 경우 바로 이웃한 데이터나 클러스터 중심까지 현재 클러스터 추정에 영향을 주지 못하는 문제점이 발생할 수 있다. ζ 의 변화에도 강인한 클러스터링 특성을 가지게 하도록 식(3)과 같이 유사 측정함수 내에 ζ 를 적용하였다. 임계치 ζ 가 클 경우 유사도 측정함수 식(3)은 더 넓은 범위까지 영향을 미치도록 하였으며 작은 경우 영향 범위를 줄이도록 구성되어 있다. 각 클러스터 중심과 데이터간의 유사도 측정을 마친 후 새로운 클러스터 중심 c_j 을 다음과 같이 결정한다[2][8].

$$c_j = \frac{\sum_{i=1}^N r_{ij} x_i}{\sum_{i=1}^N r_{ij}} \quad (4)$$

새롭게 추정된 클러스터 중심들은 데이터의 분포와 유사도 임계치 ζ 에 의하여 특정한 지역으로 수렴(convergence)된다. 클러스터 중심이 추정되면 입력차원이 d 일 때 Maximum Likelihood (ML) 평가를 위한 확률은 다음과 같이 추정한다[7][8].

$$D_{ij} = \frac{1}{(2\pi)^{d/2} \|\Sigma_j\|^{1/2}} \times \exp\left(-\frac{1}{2}(x_i - c_j)^T \Sigma_j^{-1}(x_i - c_j)\right) \quad (5)$$

식(5)에 사전확률을 곱하여 사후확률 Pw_{ij} 을 구한 후 이를 이용하여 분할행렬의 요소 u_{ij} 를 다음과 같이 계산한다.

$$u_{ij} = \frac{Pw_{ij}}{\sum_{i=1}^N Pw_{ij}} \quad (6)$$

공분산 행렬 Σ_j 는 다음과 같이 추정한다.

$$\Sigma_j = \frac{\sum_{i=1}^N u_{ij}(x_i - c_j)(x_i - c_j)^T}{\sum_{i=1}^N u_{ij}} \quad (7)$$

한번의 추정(학습)과정이 끝난 후 사전확률은 다음과 같다.

$$\Pr(x_i) = \frac{1}{N} \sum_{i=1}^N u_{ij} \quad (8)$$

초기 사전확률의 값은 전체 데이터의 수로 나눈 균등한 값을 배분한다. 식(2)에서 식(8)까지의 반복적인 학습 수행은 초기 설정된 클러스터 중심을 임계치에 의하여 정해진 클러스터의 수로 수렴하며 이들 파라미터도 최적화한다.

2.2. 클러스터링 연산량 감소 방법

하나의 클러스터 중심은 새로운 클러스터 중심을 추정하기 위하여 데이터의 수만큼 유사도를 측정해야 한다. 초기 클러스터의 수는 데이터의 수와 같으므로 데이터 집합의 크기가 큰 경우 학습이 진행되는 동안 클러스터링을 위한 연산량은 지수함수적으로 증가하는 단점을 가진다[2]. 학습이 진행되는 동안 클러스터 중심들은 특정한 공간들로 수렴하는 특성을 보이는데 이러한 특성을 이용하여 군집되는 클러스터 중심을 하나의 클러스터 중심을 통합하는 기법을 이용하여

학습에서 발생하는 연산량 문제를 해결하고자 하였다. 이 경우 데이터의 밀집 정도를 이용하여 클러스터를 추정하는 Subtractive 클러스터링 기법을 변형하여 합병규칙 및 종결조건을 새로이 하였다[2][6]. 각 클러스터 중심에서 데이터간의 유사도를 누적하여 가장 높은 누적 유사도 $D_j = \sum_{i=1}^N u_{ij}$ 를 가지는 클러스터 중심 c_j 을 새로운 대표 클러스터 c_{new} 로 다음과 같이 선택한다.

$$c_{new} = c_j \quad (9)$$

선택된 클러스터 중심과 이에 대응하는 공분산 행렬을 이용하여 다음과 같은 새로운 제거 유사도 $s_{i, new}$ 를 구한다.

$$s_{i, new} = \exp\left(-\frac{1}{2}(c_i - c_{new})^T \Sigma_{new}^{-1} (c_i - c_{new})\right) \quad (10)$$

또한 제거 유사도는 사전임계치를 이용하여 다음과 같이 다시 제한한다.

$$s_{i, j} = \begin{cases} s_{i, j}, & \text{if } s_{i, j} > \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

여기서 ϵ 은 변형된 Subtractive 클러스터링 기법에서 사용되는 제거 유사도의 범위를 제한하는 변수로써 수렴되는 클러스터의 범위를 결정한다. ϵ 의 값이 클 경우 넓은 영역에 걸쳐 클러스터 중심들이 통합되어 전체 클러스터링 학습이 빠르게 진행되는 장점을 가지는 반면 불필요한 클러스터 중심까지 수렴시킬 수 있는 문제점을 가지며 값을 작게 설정하면 위에서 발생하는 문제점은 개선할 수 있으나 학습속도 개선이 빠르게 이루어지지 않는다. 제거 유사도 범위에 속하는 클러스터 중심은 c_{new} 로 모두 수렴하는 것을 가정하고 다음 클러스터 중심 추정에서 제외한다. 이 과정이 끝나면 가장 큰 누적 유사도 D_j 을 가지는 새로운 대표 클러스터 중심을 다음과 같이 선택한다.

$$c_{new} = c_j, \text{ where } D_j = D_{new} - \sum_{i=1}^N \left(\sum_{j=1}^c s_{i, j} \right) \quad (12)$$

식(10)에서 식(12)까지의 과정을 종결조건을 만족할 때까지 반복 수행한다.

클러스터링 학습은 식(2)에서부터 학습을 시작하여 식(8)까지의 클러스터의 수 추정 및 최적화 과정을 거친 후 변형된 Subtractive 클러스터링을 마친 것을 1회 학습이라고 하고 이러한 과정을 반복한다. 그림 1에서 제안된 클러스터링의 구성을 나타내었다. 클러스터의 학습은 먼저 유사도를 측정하고 임계치를 이용하여 클러스터 추정의 허용 가능한 범위를 정한다. 여기에서 추정된 정보를 이용하여 각 클러스터 파라미터를 MLE 기법을 이용하여 최적화 한다. 하나의 클러스터 중심은 모든 데이터들과 유사도와 확률을 측정해야 하므로 데이터의 양이 많은 경우 초기 클러스터 중심의 수도 같게 설정되어 연산량이 증가한다. 모델이 학습하는 동안 과도한 연산량 문제를 해결하기 위해 변형된 Subtractive 클러스터링 기법을 적용하였다. 학습이 진행되는 동안 클러스터들은 특정한 지점으로 수렴하게 되는데 각 클러스터의 누적 밀도를 이용하여 같은 영역내의 중복된 클러스터를 하나로 통합하는 방법을 이용하여 학습 중 연산량을 줄이고자 하였다. 일련을 과정이 끝나면 종결조건을 검사하여 조건에 만족하지 않으면 그림과 같이 반복 수행을 통하여 자율적으로 클러스터의 수를 추정하고 이들 파라미터를 최적화 한다.

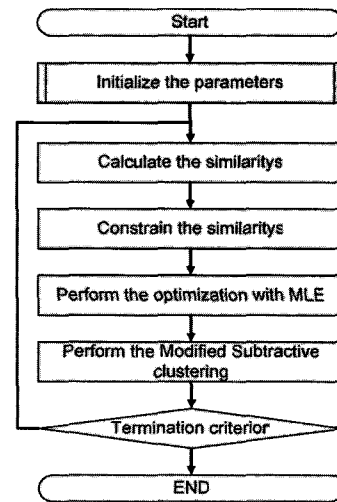


그림 1. 제안된 클러스터링 방법
Fig. 1 Proposed clustering method

3. 뉴로-퍼지 시스템의 구성

본 기법에서 이용한 뉴로-퍼지 시스템은 언어적 소속함수 형태를 가지는 전제부와 1차 선형방정식 형태를 가지는 결론부로 구성되어 있는 TSK 퍼지 모델을 이용하였다[2][5][17]. 일반적인 클러스터 모델의 경우 자신의 영역을 표현하는 방식으로 보통 2차 방정식이나 볼록(convex) 함수를 많이 이용한다. 이 특성은 퍼지 모델의 전제부 소속함수와 동일하며 클러스터링의 결과를 직접 퍼지 모델에 적용할 수 있는 장점을 가진다. TSK 퍼지 모델의 결론부는 1차 선형방정식 형태로 구성이 되어 있어 비선형 입력 공간의 선형 추론으로 고려되며 퍼지 규칙에 적용하면 다음과 같다[5].

$$\text{Rule } i: \text{ IF } X_1 \text{ is } A_i \text{ and } X_2 \text{ is } B_i, \dots \text{ THEN } f_i = p_i X_1 + q_i X_2 + \dots + r_i \quad (13)$$

여기서 $X_{1,2}$ 는 첫 번째와 두 번째 입력(차원)을 나타내며, A_i, B_i 는 각 입력 차원의 i 번째 퍼지 소속함수를 의미하며, i 번째 규칙에 의한 결론부 출력 f_i 는 각 출력 선형 방정식 파라미터 p_i, q_i, \dots, r_i 와 입력간의 곱을 의미한다. TSK 퍼지 모델은 1층에서 5층으로 나누어 해석할 수 있는데 1층에서는 수치적 입력에 대해 퍼지화를 시키며 2층에서 각 입력차원의 퍼지수를 곱하며 3층에서 2층의 출력을 정규화(Normalization)시키고 4층에서 결론부 파라미터와 연산을 실시하여 각 규칙에 의한 부분 출력을 추정한다. 5층에서는 가중 평균법을 이용한 최종 출력을 추론한다[2].

클러스터링에 의한 모델의 구성방식은 1층에서 소속함수를 할당하는 방식은 같으나 2층에서 4층까지 규칙의 연결은 입력차원에 따른 각 클러스터에 속하는 소속함수만으로 이루어지며 개념을 그림 2에 나타내었다. 4층에서의 출력이 각 클러스터에 의한 추정 결과를 나타낸 것이며 5층은 각 규칙들에 의한 추론을 가중평균한 것이다. 따라서 각 클러스터에 의한 퍼지 모델의 영향은 4층에서 볼 수 있다.

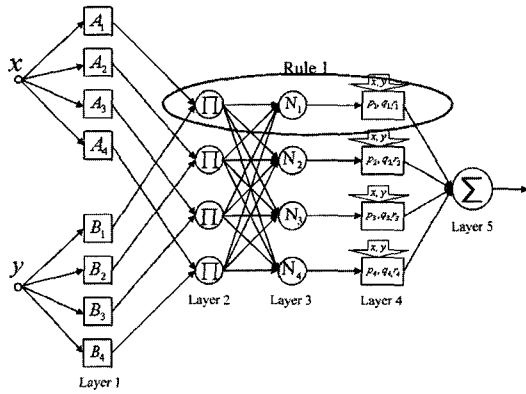


그림 2. 뉴로-퍼지 모델
Fig. 2 Neuro-fuzzy model

제안된 기법에서는 클러스터링에 의하여 나온 결과를 뉴로-퍼지 모델을 이용하여 평가를 하고 이 결과를 다시 클러스터 학습에 영향을 주게 하였다. 먼저 j 번째 규칙에 대한 4층의 출력 $\bar{\omega}_j f_j$ 는 다음과 같다.

$$o_j = \bar{\omega}_j f_j = \bar{\omega}_j (p_j X_1 + q_j X_2 + \dots + z_j) \quad (14)$$

식(13)의 출력과 사전에 주어진 출력을 이용하여 결론부 오차 $e_{j,NF}$ 를 다음과 같이 추정할 수 있다.

$$e_{j,NF} = y_i - \bar{\omega}_j f_j, \text{ for } j=1, \dots, m \quad (15)$$

여기서 y_i 는 i 번째 규칙에 의한 결론부의 출력을 나타낸다. 식(15)를 이용하여 결론부에서 오차에 대한 소속함수 μ_{e_i} 를 다음과 같이 생성할 수 있다.

$$\mu_{e_i} = \exp\left(-\frac{1}{2} \frac{(e_{j,NF} - \mu_{ei})^2}{\sigma_{ei}^2}\right) \quad (16)$$

여기서 μ_{ei} 와 σ_{ei}^2 는 결론부의 오차 소속함수의 중심과 분산을 나타낸다. 이를 이용하여 클러스터링의 유사도 측정함수에 영향을 다음과 같이 준다.

$$r_{ij} = r_{ij} \times (\mu_{e_i} \cdot (1 - \alpha) + \alpha) \quad (17)$$

여기서 결론부 변수 α 는 뉴로-퍼지 모델의 부분 출력에 대한 영향이 클러스터링에 미치는 영향범위를 나타내며 0에서 1사이의 값을 가진다.

클러스터링과 뉴로-퍼지 모델의 학습기법은 다음과 같다. 클러스터링에 의하여 추정된 결과는 뉴로-퍼지 모델의 전제부 소속함수의 파라미터로 이용되고 결론부 파라미터는 최소자승법에 의하여 추정된다[2]. 생성된 모델은 각 부분 출력을 추정하게 되며 여기서 발생된 오차는 다시 클러스터 학습에 이용되는 형태로 반복 수행하게 된다. 결론부의 경우 선형방정식 형태로 구성되어 있어 파라미터를 추정하는 방식은 선형 시스템을 만족하는 다양한 기법들이 제안되어 있다. 제안된 기법에 의해 구성된 뉴로-퍼지 모델의 학습 진행을 그림 3에 나타내었다. 1회 클러스터링을 실시한 후 추정된 파라미터를 이용해 뉴로-퍼지 모델을 구성하고, 이를 이용해 생성된 오차를 이용하여 다시 클러스터링을 학습하는 클러스터링 기반 학습을 실시하였다.

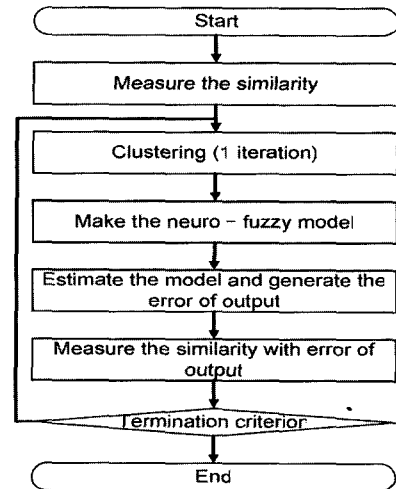


그림 3. 제안된 뉴로-퍼지 학습 모델
Fig. 3 Proposed neuro-fuzzy model for learning

TSK 퍼지 모델은 언어적 입력을 가지는 전제부와 방정식 형태의 결론부를 가지고 있다. 제안된 기법에서는 전제부 소속함수를 Gaussian 퍼지 소속함수로 이용하여 전제부 학습에 필요한 요소는 중심 μ 와 분산 σ 로 하였으며, 전제부 학습에 의하여 선형방정식 형태인 결론부를 선형 파라미터 추정기법인 최소자승법으로 전제부 학습에 의존한 추정방식을 이용하였다. 전제부 소속함수의 학습과 동시에 결론부 학습이 이루어진다. 전제부 소속함수에 필요한 학습은 클러스터링에서 추정된 중심과 분산을 이용하여 실시하며 학습된 모델의 출력에서 추정된 오차정보를 모델의 전제부가 아닌 클러스터링의 학습에 적용하였다. 이를 정리하면 다음과 같다. 먼저 클러스터링을 실시하여 클러스터의 중심과 분산을 추정한다. 추정된 중심과 분산을 뉴로-퍼지 모델의 전제부 Gaussian 퍼지 소속함수의 파라미터에 대입하고 최소자승법을 이용하여 결론부 선형 방정식 파라미터를 추정하는 형태로 전체 모델을 구성한다. 실제 주어진 출력과 모델의 출력을 비교하여 오차 정보를 획득하고 이를 다시 뉴로-퍼지 모델이 아닌 클러스터링 학습에 적용하여 클러스터링을 실시한다. 이러한 반복과정을 정해진 학습 횟수나 조건에 따라 반복하여 학습을 실시한다.

4. 시뮬레이션 및 결과

본 논문에서는 대표적인 benchmark 문제인 Box-Jenkins의 가스로(gas furnace) 데이터에 적용하여 제안된 기법의 유용성을 보이고자 한다[2][4]. 이 예제는 전형적인 비선형 시계열 데이터로써 하나의 입력과 하나의 출력으로 구성된 시계열 데이터이다. 입력으로 매탄 흐름률과 출력으로 이산화탄소 CO_2 의 비율로 $y(k-i), i=1,2,3,4, u(k-j), j=1,2, \dots, 6$ 으로 입출력 296개의 데이터 쌍으로 구성되어 있고, Jang의 입력선택방법에 의해 입출력 데이터 쌍은 다음과 같이 선택하였다.

$$y(k+1) = f(y(k), u(k-3)) \quad (18)$$

입출력 데이터의 홀수 열을 전체 모델의 학습에 이용하였고 짝수 열을 모델 성능을 검증하는데 이용하였다. 사전 임계치 ζ 와 결론부 변수 α 를 가변하였을 때 클러스터 수의 변화를 표 1에 나타내었다. 여기서 ■은 학습이 진행되지 않은 영역을 말한다. 클러스터링의 조건을 변화시켜 수행했을 때 클러스터의 수가 변하는 것은 유사도 측정함수의 유사도 인 정범위가 다르게 되는 것과 결론부 변수의 크기에 의해 오차가 클러스터에 미치는 영향이 달라지기 때문이다. 표에서 알 수 있듯이 사전임계치가 커지거나 뉴로-퍼지 모델 출력의 오차 영향이 커질수록 규칙의 수가 작아지며 그 반대의 경우 많아지는 것을 볼 수 있다. 또한 모델 출력이 클러스터링 학습에 영향을 크게 미치는 경우($\alpha=0.7$, 출력영향 70%, 입력의 영향 30%) 모델의 학습이 되지 않은 영역이 발생하는 것을 알 수 있다. 일반적인 클러스터링은 입력공간에 대하여 실시하는 것이나 제안된 기법에서는 출력공간의 정보를 추가하였다. 그러나 출력 공간의 정보가 너무 큰 경우 모델 학습에 문제가 발생할 수 있다.

표 1. 학습에 따른 규칙의 수
Table 1. Rules following learning model application

$\zeta \backslash \alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.05	8	13	11	10	9	11	4
0.06	14	12	12	9	10	7	6
0.07	14	14	11	14	9	7	6
0.08	14	13	12	10	12	8	5
0.09	12	12	10	13	9	8	
0.1	13	14	10	9	9	7	
0.13	13	11	6	9	6	6	
0.15	11	8	9	7	7	4	
0.17	10	8	9	5	6	5	
0.2	8	7	6	5	4	4	3
0.25	2	4	3	3	4	4	

사전 임계치와 결론부 변수를 각각 0.2와 0.4로 한 후 제안된 방법을 수행하였을 때 추정된 클러스터 분포를 그림 4에 나타내었다.

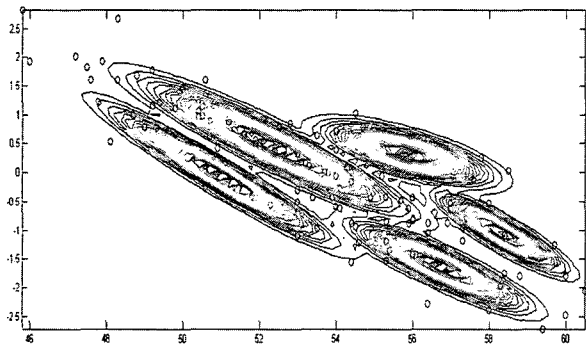


그림 4. 입력과 클러스터링 결과
Fig. 4 Inputs and clustering result

전체 모델의 학습이 진행되는 동안 클러스터의 수는 사전 임계치와 결론부 변수에 의하여 점차 수렴한다. 학습의 시작에서 클러스터의 수는 전체 데이터의 수와 같다. 따라서 각

클러스터와 데이터 간의 연산량은 $N \times N$ 으로 과대하게 많다. 또한 클러스터 파라미터를 최적화하기 위하여 다시 $N \times N$ 의 연산을 실시하여야 한다. 데이터가 큰 경우 전체 모델의 최적화를 위하여 소요되는 학습시간은 크게 증가할 수밖에 없다. 이러한 문제점을 해결하기 위하여 제안된 기법은 변형된 Subtractive 클러스터링을 이용하여 동일한 수렴방향으로 향하는 유사한 클러스터를 제거하여 연산량을 줄이고자 하였다. 학습이 진행되는 동안 사전 임계치와 결론부 변수를 가변하였을 때 결과를 그림 5에 나타내었다. 그림에서 보면 사전 임계치가 큰 경우 초기 학습에서 클러스터의 수가 크게 감소하는 것을 볼 수 있다.

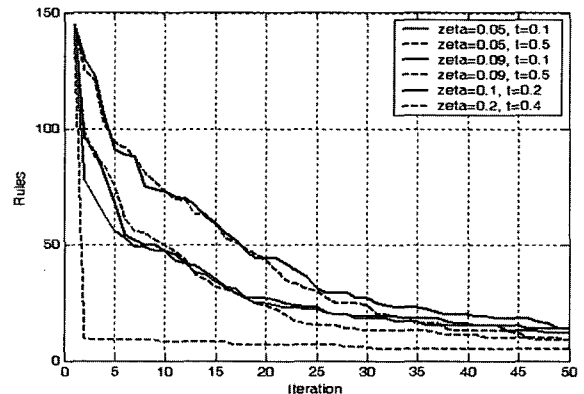


그림 5. 학습에서의 규칙 수의 변화
Fig. 5 Rule variation in learning process

학습데이터에 의하여 모델 오차의 변화를 Root Mean Square Error (RMSE)를 이용하여 표 2에 나타내었다. 학습 데이터에 대한 오차를 비교하면 제안된 모델의 성능이 일정한 범위 내에서 유사한 것을 알 수 있다. 학습된 모델을 검증 데이터를 이용하여 평가하였을 때 결과를 표 3에 나타내었다. 표 2의 결과에서와 같이 검증데이터에 의한 모델 오차에서 볼 수 있듯이 제안된 기법에 의한 모델 학습 방법이 넓은 영역에 걸쳐서 좋은 성능을 보이는 것을 알 수 있다. 실험 결과를 기존의 연구 결과와 비교하면 표4와 같다.

표 2. 학습데이터에 의한 모델 오차
Table 2. RMSE of training data

$\zeta \backslash \alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.05	0.129	0.099	0.133	0.136	0.141	0.120	0.149
0.06	0.098	0.113	0.121	0.133	0.139	0.139	0.123
0.07	0.097	0.109	0.124	0.096	0.143	0.126	0.139
0.08	0.097	0.121	0.096	0.128	0.105	0.122	0.124
0.09	0.102	0.121	0.135	0.119	0.126	0.157	
0.1	0.116	0.104	0.108	0.128	0.125	0.129	
0.13	0.116	0.125	0.157	0.145	0.118	0.126	
0.15	0.131	0.137	0.137	0.133	0.129	0.116	
0.17	0.121	0.153	0.125	0.129	0.129	0.129	
0.2	0.129	0.151	0.127	0.132	0.153	0.136	0.165
0.25	0.168	0.167	0.141	0.138	0.136	0.131	

표 4에서 볼 수 있듯이 제안된 기법에 의해 학습된 모델이

기존의 연구 결과와 비교할 때 학습데이터에 대한 오차뿐만 아니라 검증데이터에 대한 오차도 일정한 범위 내에서 감소하는 것을 볼 수 있다. 특히 모델이 과적합 학습(overfitting)이 되는 경우 학습데이터에 대한 성능은 좋은 반면 검증데이터에 대해 적응을 하지 못하는 문제가 발생한다. 이 경우 생성된 모델이 주어진 시스템을 정확하게 표현하지 못하는 문제점을 가진다. 이런 관점에서 제안된 기법은 학습데이터에 대한 오차뿐만 아니라 검증데이터에 대한 오차 성능 역시 주어진 조건을 만족한다고 할 수 있다. ζ 를 0.1, α 를 0.2로 설정하고 각 입력의 최소 및 최대 범위를 $y(k)$ 는 [45.8, 60.5]로 정하고 $u(k-3)$ 는 [-2.716, 2.812]로 하였으며 출력 선형 방정식의 최소 및 최대 범위를 [45.6, 60.4]로 하였을 때 실험에 이용한 TSK 퍼지 모델의 전제부 소속함수의 Gaussian 소속함수의 파라미터와 결론부 선형방정식의 파라미터를 표 5에 나타내었다.

표 3. 검증데이터에 의한 모델 오차
Table 3. RMSE of checking data

$\xi \backslash \alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7
0.05	0.176	0.224	0.167	0.169	0.207	0.215	0.187
0.06	0.208	0.987	0.233	0.218	0.163	0.231	0.189
0.07	79.25	0.169	0.215	0.296	0.173	0.177	0.199
0.08	0.151	0.219	0.192	0.302	0.315	0.152	0.154
0.09	0.221	0.201	920	2.361	0.434	0.15	
0.1	0.362	0.169	1.535	0.186	0.189	0.207	
0.13	0.183	0.171	0.171	0.269	0.401	0.213	
0.15	0.156	0.200	0.168	0.175	0.162	0.177	
0.17	0.170	0.187	0.163	0.155	0.435	0.182	
0.2	0.176	0.175	0.153	0.186	0.175	0.189	0.176
0.25	0.181	0.203	0.191	0.199	0.194	0.209	

표 4. 기존의 연구결과와 비교

Table 4. Comparison of results with previous ones

Error(RMSE) Method	Number of rules	Training error	Checking error	Remark
Pedrycz[11]	81	0.320	None	Training only
Xu[12]	25	0.328	None	Training only
Sugenol[13]	2	0.359	None	Training only
Abonyi[14]	16	0.154	None	Training only
Babuska[15]	23	0.124	None	Training only
Jang[16]	4	0.13	0.53	
Oh[17]	4	0.026	0.272	
Proposed $\zeta=0.1, \alpha=0.4$	9	0.128	0.186	50 iter
Proposed $\zeta=0.2, \alpha=0.4$	5	0.132	0.186	50 iter
Proposed $\zeta=0.1, \alpha=0.2$	14	0.104	0.169	50 iter

표 5. TSK 퍼지 모델의 파라미터
Table 5. Parameters in the TSK fuzzy model

Num of Rule	Antecedent Gaussian MF (σ_i, μ_i)	Consequent Linear Eq(p_i, q_i, r_i)
1	$y(k) : 0.938, 50.8$ $u(k-3) : 0.318, 1.118$	0.6943, -0.8409, 16.16
2	$y(k) : 1.02, 52.5$ $u(k-3) : 0.009424, 0.8334$	0.9729, 0.1327, 0.07617
3	$y(k) : 1.287, 53.71$ $u(k-3) : 0.2404, -0.6181$	-0.009401, -2.048, 53.55
4	$y(k) : 0.8143, 58.46$ $u(k-3) : 0.12, -1.855$	0.3187, 0.4728, 40.81
5	$y(k) : 4.547, 52.28$ $u(k-3) : 0.5254, -0.8475$	0.9112, -0.1494, 5.672
6	$y(k) : 1.104, 57.18$ $u(k-3) : 0.1738, -1.913$	0.6499, -2.332, 16.19
7	$y(k) : 2.485, 57.65$ $u(k-3) : 0.3596, -0.9358$	0.6718, -1.151, 16.91
8	$y(k) : 0.09566, 57.97$ $u(k-3) : 0.01746, 0.04999$	14.95, 0.008691, 0.2556
9	$y(k) : 2.026, 49.19$ $u(k-3) : 0.3059, 1.365$	0.5347, -1.14, 24.25
10	$y(k) : 2.05, 49.76$ $u(k-3) : 0.3715, 0.4899$	0.4914, -1.618, 26.94
11	$y(k) : 1.797, 54.84$ $u(k-3) : 0.1211, 0.2228$	1.701, 14.27, -42.61
12	$y(k) : 1.913, 53.74$ $u(k-3) : 0.175, 0.03022$	0.6453, 5.879, 19.39
13	$y(k) : 1.229, 56.06$ $u(k-3) : 0.2812, 0.2121$	0.9244, 0.6165, 2.809
14	$y(k) : .408, 57.02$ $u(k-3) : 0.01599, -0.9694$	-2.715, -29.46, 181.5

5. 결 론

본 논문에서는 클러스터링을 기반으로 한 뉴로-퍼지 모델링을 실시하였다. 클러스터링에서는 사전 임계치와 결론부 변수를 이용하여 자율적으로 클러스터의 수를 추정하고 이들 파라미터를 최적화하는 기법을 제안하였으며 이를 뉴로-퍼지 모델의 학습에 적용하여 모델의 성능을 개선하고자 하였다. 또한 뉴로-퍼지 모델의 부분 출력을 이용하여 전형적인 비교사 학습인 클러스터링을 교사 학습으로 확장하여 전체 성능을 개선하고자 하였다. 학습과정에서 발생하는 연산량 문제를 수렴되는 클러스터의 분포를 이용하여 대표 클러스터를 선택하는 기법을 이용하여 전체 학습 연산량을 개선하였다. 향후 연구과제로는 제안된 기법이 가지는 장점에서 불구하고 여전히 학습 초기단계에서 발생하는 과다 연산량 문제를 개선하는 것과 특정한 영역에서 과적합 학습이 발생하는 문제 등을 해결하는 것 등이 있다

참 고 문 헌

[1] C. T. Lin, C. S. G. Lee, "Neural fuzzy Systems : A Neuro-Fuzzy Synergism to Intelligent Systems",

Prentice Hall, 1996.

[2] J. S. R. Jang, C. T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence", Prentice Hall, 1997.

[3] Kazuo Tanaka and Hua O. Wang, Fuzzy Control Systems Design and Analysis : A Linear Matrix Inequality Approach, John Wiley & Sons, 2001.

[4] 김승석, 광근창, 유정웅, 전명근, "계층적 클러스터링과 Gaussian Mixture Model을 이용한 뉴로-퍼지 모델링," 한국퍼지및지능시스템학회 논문지, Vol. 13, No. 5, pp. 512-519, 2003.

[5] J. S. R. Jang, "ANFIS : Adaptive Network-based Fuzzy Inference System", IEEE Trans on System, Man, and Cybernetics, Vol.23, No. 3, pp. 665-685, 1993.

[6] R. R. Yager and D. P. Filev, "Generation of Fuzzy Rules by Mountain Clustering", Journal of Intelligent and Fuzzy System, Vol.2, pp. 209-219, 1994.

[7] G. Xuan, W. Zhang, and P. Chai, "EM algorithm of Gaussian Mixture Model and Hidden Markov Model", International Conference on Image Processing Proceeding, Vol. 1, pp. 145-148. 2001.

[8] Jeff Bilms, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", ICSI TR-97-021, April 1998.

[9] Ching-Chang Wong, Chia-Chong Chen, and Mu-Chun Su, "A novel algorithm for data clustering", Pattern Recognition, Vol. 34, Issue. 2, pp. 425-442, 2001.

[10] Ching-Chang Wong and Chia-Chong Chen, "A Hybrid Clustering and Gradient Descent Approach for Fuzzy Modeling", IEEE Trans on Systems, Man, and Cybernetics-Part B : Cybernetics, Vol. 29, No.6, pp. 686-693, 1999.

[11] Witold Pedrycz, "An Identification Algorithm in Fuzzy Relational Systems", Fuzzy Sets and Systems, Vol. 13, No. 2, pp. 153-167, 1984.

[12] C. Xu and L. Lu, "Fuzzy model Identification and Self-Learning for Dynamic Systems", IEEE Trans on Systems, Man and Cybernetics, Vol. SMC-17, pp. 683-689, 1987.

[13] M. Sugeno and K. Tanaka, "Successive Identification of a Fuzzy Model and Its Application to Prediction of a Complex System", Fuzzy Sets and Systems, Vol. 42, pp. 315-334, 1991.

[14] J. Abonyi, L. Nagy, and F. Szeifert, "Adaptive Fuzzy Inference Systems and Its Application in Modeling Based Control", Chemical Engineering Research and Design, Trans IChemE, Vol. 77A, pp. 281-290, 1999.

[15] Janos Abonyi, Robert Babuska, and Ferenc Szeifert, "Fuzzy Modeling With Multivariate Membership Functions : Gray-Box Identification and Control Design", IEEE Trans on Systems,

Man, and Cybernetics-Part B : Cybernetics, Vol. 31, No. 5, pp. 755-767, 2001.

[16] S. R. Jang, "Input Selection for ANFIS Learning", Proceeding of Fifth IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 8-11, 1996.

[17] S. K. Oh and Witold Pedrycz, "Identification of Fuzzy System by Means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems", Fuzzy Sets and Systems, Vol. 115, No. 2, pp. 205-230, 2000.

[18] 김승석, 김성수, 유정웅, "새로운 클러스터링 알고리즘을 적용한 향상된 뉴로-퍼지 모델링", 대한전기학회 논문지, Vol. 53D, No. 7, pp. 536-543, 2004.

저자 소개



김승석(Sung-Suk Kim)

1998. 2 : 충주대학교 전기공학과(공학사)
 2002. 2 : 충북대학교 전기공학과(공학석사)
 2005. 8 : 충북대학교 전기공학과(공학박사)
 2005. 1~현재 : 충북대학교 컴퓨터 정보통신연구소 전임 연구원
 2005. 3~현재 : 충북대학교 전전컴 초빙전임강사

관심분야 : 뉴로-퍼지 모델링, 지능모델 학습, 지능제어
 E-mail : powerkim@chungbuk.ac.kr



유정웅(Jeong-Woong Ryu)

1965. 2 : 한양대학교 전기공학과(공학사)
 1976. 2 : 단국대학교 전자공학과(공학석사)
 1987. 2 : 단국대학교 전자공학과(공학박사)
 1979. 3~현재 : 충북대학교 전기공학과 교수

관심분야 : 가변구조제어, 지능제어, QFT
 E-mail : jwryu@chungbuk.ac.kr



김용태(Yong-Tae Kim)

1991년 : 연세대학교 전자공학과(학사)
 1993년 : KAIST 전기 및 전자공학과 졸업(공학석사)
 1998년 : KAIST 전기 및 전자공학과 졸업(공학박사)
 1998년~2000년 : (주)삼성전자
 2000년~2001년 : (주)네오다임소프트
 2002년~현재 국립 한경대학교 정보제어공학과 조교수

관심분야 : 지능로봇, 지능시스템, 지능제어, 학습제어
 E-mail : ytkim@hknu.ac.kr