Proceedings of KFIS Autumn Conference 2007 Vol. 17, No. 2.

상호 노드 정보를 이용한 클러스터 기반 퍼지 모델트리

Cluster Based Fuzzy Model Tree using Node Information

박진일[1], 이대종[2], 김용삼[1], 전명근[1]

[1] 충북대학교 전기전자컴퓨터공학부
E-mail: moralskr@yahoo.co.kr
[2] 충북대학교 BK21 충북정보기술사업단
E-mail: djmidori@empal.com

요 약

본 논문에서는 기존의 클러스터 기반 퍼지 모델트리에서 트리의 깊이에 따른 over-fitting으로 인한 훈련 및 검증데이터의 일관성 문제점을 해결하기 위해 상호 노드간의 정보를 고려하는 방법을 제안하고자 한다. 제안된 방법은 우선 입력과 출력변수의 속성을 고려한 퍼지 클러스터링에 의해 중심벡터를 계산한 후, 중심벡터들과 입력 속성간의 소속도를 이용하여 구간 분할된 영역별로 각각의 선형모델을 구축한다. 예측 단계에서는 입력된 데이터가 잎노드에 도달하는 노드간의 중심벡터와 입력 데이터간의 거리값에 따른 소속도를 계산한 후 최종적으로 무게 중심법을 이용하여 출력값을 예측하게 된다. 제안된 방법의 우수성을 보이기 위해 다양한 벤치마크 데이터를 대상을 실험한 결과, 기존의 클러스터 기반 퍼지 모델트리보다 향상된 성능을 보임을 알 수 있었다.

Key Words : 데이터 예측, 퍼지 클러스터, 퍼지 모델트리

1. 서 론

일반적으로 예측문제에서는 연속적인 입력변수 및 출력값을 갖는 데이터들이 대부분을 차지한다. 결정트리의 하나의 분류인 회귀트리는 말단 노드에 위치한 잎노드(leaf node)에 속한 연속적인 출력값의 평균값을 계산함으로 예측력의 저하를 초래한다. 이러한 문제점을 해결하기 위해 모델트리 기반의 다양한 알고리즘이 제안되고 있으며[1], 주된 기법으로 M5[2], RETIS[3], M5'[4], RegTree[5] 및 HTL[6] 등이 있다. 모델트리는 회귀트리와 구조적으로 동일하지만 말단의 잎노드에 속한 출력값의 평균값을 계산하는 회귀트리와 달리 연속적인 입력값과 출력값을 이용하여 예측 오차값이 최소화되는 계수값을 계산한 후, 계산된 계수값을 이용하여 출력값을 예측한다. 이러한 모델트리도 회귀트리와 같이 데이터를 반복적으로 분리하여 트리구조를 생성하는 상-하 추론 모델트리(TIMIT:Top-down Induction of Model Tree) 형식을 갖는다.

클러스터 기반 퍼지 모델트리(c-fuzzy model tree)는 다중 입력변수들 중 중요한 특성을 갖는 변수를 선정한 후 분리기준인 SDR값을 이용하여 입력공간을 분할하는 모델트리 방식과 다르게 모든 입력속성들을 고려하여 분리기준을 판정하는 방법이다. 이 방법은 모든 입력속성을 고려하여 퍼지 클러스터에 의해 계산된 중심벡터를 설정한 후, 각각의 중심벡터들과 입력속성간의 소속도를 이용하여 내부 노드를 형성하고, 형성된 내부노드에서 각각의 선형모델을 구축한다. 노드의 분리기준으로서 부모노드(parent node)에서 구축된 모델에서 계산된 오차값이 자식노드(child node)에서 계산된 오차값보다 클 경우에 분기가 이루어진다. 최종 단계에서는 임의의 입력데이터와 잎노드에서 계산된 클러스터 중심값과 비교하여 소속도가 높은 클러스터에 속한 선형모델을 선택하여 출력값을 예측한다[7,8].

논문에서는 클러스터 기반 퍼지 모델트리에서 과도한 훈련으로 인하여 발생하는 over-fitting 등으로 인한 검증데이터의 일관성 문제를 극복하기 위한 방법을 제안하고자 한다. 클러스터 기반 퍼지 모델트리는 예측 단계에서 잎노드의 선형모델만을 고려하지만, 본 논문에서는 입력된 데이터가 모델트리의 상위노드에서 잎노드에 도달하는 노드간의 중심벡터와 입력 데이터간의 거리값에 따른 소속도를 계산한 후 최종적으로 무게 중심법을 이용하여 출력값을 예측하는 방법을 제안하고자 한다.

2. 상호 노드 정보를 이용한 클러스터 기반 퍼지모델트리

일반적으로 FCM(Fuzzy C-Means) 알고리즘은 입력변수 X만을 고려하지만, 모델트리 알고리즘은 출력값을 포함하여 데이터의 특성이 반영되도록 입력과 출력을 포함한 Z를 이용하여 중심벡터를 구한다[8]. 따라서 FCM에 의해 입력패턴에 대하여 계산된 i번째 중심벡터 $v(i) = [v_1(i), v_2(i), \cdots, v_q(i)]$ 와 출력패턴에 대하여 계산된 i번째 중심벡터 $w_i = v_{(q+1)}(i)$를 얻을 수 있다. 중심벡터를 구한 후 하위 노드로 분기할 것인지의 판정은 다음 네 가지의 조건을 고려한다.

표 1. 분기조건

- 분기 전 예측 오차값이 설정된 값 (S_1)이상일 때
- 분기 후 모든 클러스터에 포함되는 데이터의 개수가 설정된 값 (S_2)이상일 때
- 분기 전과 분기 후의 오차값 향상이 설정된 값 (S_3)이상 일 때
- 분기된 트리의 깊이 (depth)가 설정된 값 (S_4)이하 일 때

앞서 기술된 표기 방법에 따라 클러스터 기반 퍼지 모델트리를 이용하여 데이터 모델을 구하는 과정을 단계별로 설명하면 다음과 같다.

[단계 1] 표 1에 언급된 분기조건에 적용되는 값 S_1, S_2, S_3, S_4을 설정한다.

[단계 2] 모델트리의 특정 노드에 존재하는 $h(h \geq S_2)$ 개의 입출력 데이터 $\{X, Y\} \in R^{q \times h}$에 대하여 최소자승(LSE:Least Square Error)법을 이용하여 선형 계수값을 구한 후, 실제 출력값과 예측값과의 오차값을 다음과 같이 산출한다. 식 (2)로부터 구한 오차값 E_b값이 S_1이상일 때 다음 단계를 실행하고, 그렇지 않을 경우 분기를 정지한다.

$$\hat{y}(k) = a_1 \cdot x_1(k) + a_2 x_2(k) + \cdots, + a_q x_q(k) + a_{q+1} \quad (1)$$

for $k = 1, 2, \cdots, h$

$$E_b = \sqrt{\sum_{k=1}^{h} (\hat{y}(k) - y(k))^2 / h} \quad (2)$$

[단계 3] FCM 알고리즘을 이용하여 [단계 1]의 노드에 존재하는 입출력 데이터를 이용하여 c개의 클러스터 중심값을 산출한 후, 다음과 같이 입력값을 c개의 중심값 중에서 소속도가 높은 클러스터로 하위노드 X_i의 입출력 클러스터를 형성한다.

$$\begin{cases} X_i = \{x(k) \mid u_i(x(k)) > u_j(x(k))\}, \ all \ i \neq j \\ Y_i = \{y(k) \mid (x(k)) \in X_i\} \end{cases} \quad (3)$$

여기서, U_i는 아래와 같이 상위노드에 있는 데이터와 i번째 하위노드의 중심벡터에 대한 소속값을 나타낸다.

$$U_i = [\ u_i(x(1)), u_i(x(2)), \cdots, u_i(x(h))\ ] \quad (4)$$

[단계 4] 각각의 하위노드인 $X_1, X_2, \cdots, X_c$에 존재하는 데이터의 개수 $n_1, n_2, \cdots, n_c$를 계산한 후, 각각의 데이터의 개수중 하나라도 설정된 개수(S_2) 이하이면 분기를 정지하고 상위노드를 말단의 잎노드로 간주한다. 그렇지 않을 경우 [단계 5]를 실행한다.

[단계 5] 하위노드 중 클러스터 i에 해당하는 입출력 데이터 $\{X_i, Y_i\}$만을 이용하여 [단계 1]에서 계산된 방법과 마찬가지로 실제 출력값과 예측값과의 오차값을 각각 산출한 후, 하위노드에 존재하는 모든 데이터를 이용하여 에러값 E_f를 구한다.

$$\hat{y}_i(k) = b_{i1} \cdot x_{i1}(k) + b_{i2} x_{i2}(k) + \cdots, + b_{iq} x_{iq}(k) + b_{iq+1} \quad (5)$$

for $k = 1, 2, \cdots, n_i$

$$E_f = \sqrt{\left(\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\hat{y}_i(j) - {}_i y(j))^2\right) / \left\{\sum_{i=1}^{c} n_i\right\}} \quad (6)$$

여기서, E_f은 모든 클러스터에 해당되는 데이터들을 이용하여 예측된 출력값과 실제 출력값과의 오차값을 나타낸다.

분기전 상위노드에서 식 (2)를 이용하여 계산된 오차값 E_b와 분기 후 모든 하위노드에서 계산된 에러값 E_c간의 차 $\delta = E_b - E_f$를 계산한 후, δ값이 증가하거나 아주 적은 값을 갖는 임계값 (S_3)이하의 값을 가질 경우 분기과정을 정지한다. 즉, δ가 증가한다는 의미는 분기를 하였음에도 불구하고 오차값이 증가함을 의미하고 또한 δ가 임계값 이하로 감소하지 않는다는 의미는 분기를 했음에도 오차 측면에서는 큰 효과가 없음을 의미한다.

[단계 6] 표 1의 분기조건을 만족하는 하위 노드를 대상으로 분기를 시작하며, 그 과정은

[단계 1]~[단계 5] 과정을 반복한다. 단, 트리의 깊이(depth)가 설정된 값 (S_4)를 초과할 경우 분기는 정지한다.

클러스터 기반 퍼지 모델트리에서 훈련 데이터에 대한 오차는 노드의 깊이가 증가할수록 일반적으로 감소하게 된다. 그러나 훈련데이터만을 고려하는 over-fitting등으로 인하여 검증데이터에 대한 오차가 증가하는 결과를 초래할 수 있다.

제안된 알고리즘에서는 모델트리 구축에서 FCM 알고리즘에 의한 노드의 중심벡터인 $V=[v(1), v(2), \cdots, v(c)] \in R^{q \times c}$을 고려한다.

기존의 방법에서는 c개의 클러스터 중심값 중에서 소속도가 높은 클러스터의 하위노드에서의 선형모델을 이용하였다.

입력 데이터 x_i가 구축된 모델트리에서 잎노드까지 도달하는 중심벡터가 $V_{x_i}=[v^1_{x_i}, v^2_{x_i}, \cdots, v^d_{x_i}]$이고, 각각의 노드에서의 선형모델을 다음과 같이 정의할 때

$$\begin{aligned}
\hat{y}(v^1_{x_i}) &= b_1^{v^1_{x_i}} \cdot x_1^{v^1_{x_i}}(k) + , \cdots, + b_q^{v^1_{x_i}} \cdot x_q^{v^1_{x_i}}(k) + b_{q+1} \\
\hat{y}(v^2_{x_i}) &= b_1^{v^2_{x_i}} \cdot x_1^{v^2_{x_i}}(k) + , \cdots, + b_q^{v^2_{x_i}} \cdot x_q^{v^2_{x_i}}(k) + b_{q+1} \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\hat{y}(v^d_{x_i}) &= b_1^{v^d_{x_i}} \cdot x_1^{v^d_{x_i}}(k) + , \cdots, + b_q^{v^d_{x_i}} \cdot x_q^{v^d_{x_i}}(k) + b_{q+1} \\
&\text{for } k=1,2,\cdots,q
\end{aligned} \tag{7}$$

중심벡터 V_{x_i}와 입력 데이터 x_i와의 거리값 d_i에 따른 소속도를 아래의 식 (8)을 이용하여 계산한다.

$$\mu^{v^d_{x_i}}_{x_i} = \left[\sum_{k=1}^{d} \left(\frac{d_{ik}}{d_{x_i d}} \right)^{2/(m-1)} \right]^{-1} \tag{8}$$

여기서, m은 퍼지화 정도를 나타내는 퍼지 수로써 일반적으로 2를 이용한다. 또한, $d_{x_i d}$는 p차원을 갖는 입력 데이터 x_i와 d번째 대표 중심값 $v^d_{x_i}$와의 유클리디안 거리값을 의미한다.

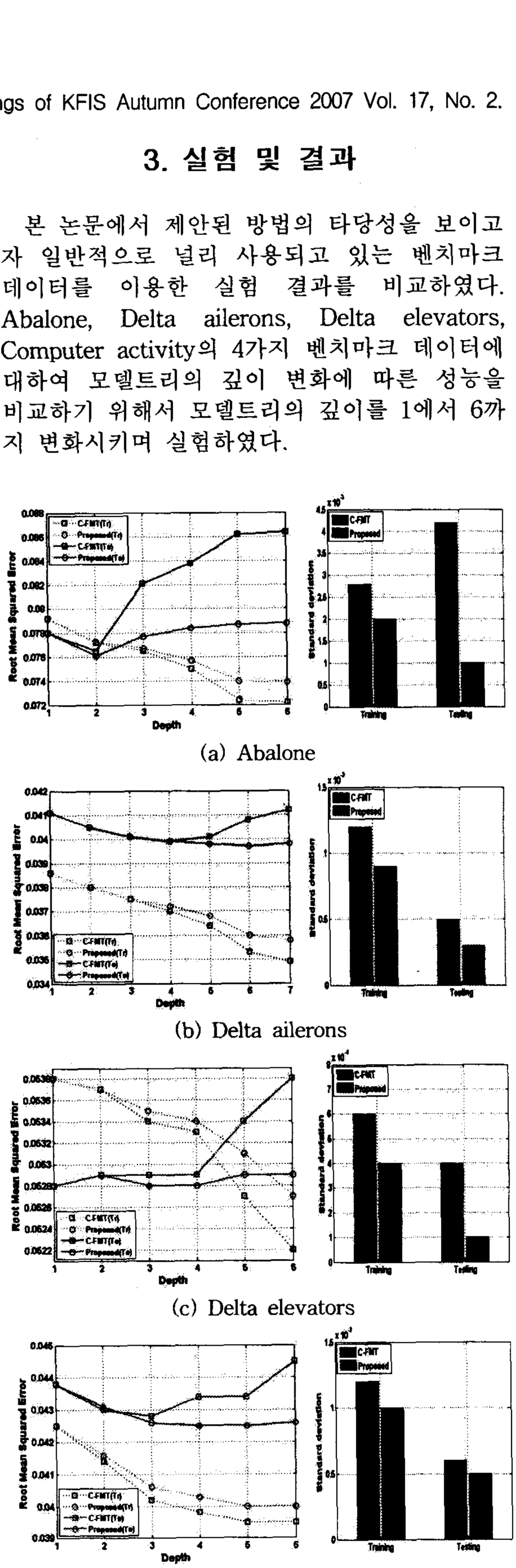
$$d_{x_i d} = d(x_i, v^d_{x_i}) = \left[\sum_{k=1}^{p} (x_{ik} - v^d_{x_i})^2 \right]^{-1} \tag{9}$$

최종 출력 Y_{x_i}는 퍼지 비퍼지화 방법으로 사용되고 있는 무게 중심법을 이용하여 다음과 같이 얻어진다.

$$Y_{x_i} = \frac{\sum_{k=1}^{d} \hat{y}(v^k_{x_i}) \cdot u^{v^k_{x_i}}_{x_i}}{\sum_{k=1}^{d} u^{v^k_{x_i}}_{x_i}} \tag{10}$$

3. 실험 및 결과

본 논문에서 제안된 방법의 타당성을 보이고자 일반적으로 널리 사용되고 있는 벤치마크 데이터를 이용한 실험 결과를 비교하였다. Abalone, Delta ailerons, Delta elevators, Computer activity의 4가지 벤치마크 데이터에 대하여 모델트리의 깊이 변화에 따른 성능을 비교하기 위해서 모델트리의 깊이를 1에서 6까지 변화시키며 실험하였다.



(a) Abalone

(b) Delta ailerons

(c) Delta elevators

(d) Computer activity

그림 1. 모델 트리의 깊이 변화에 따른 특성 비교

그림 1에서는 각각의 벤치마크 데이터에서 모델트리의 깊이 변화에 따른 특성들을 보여주고 있다. 그림 1에서 알 수 있듯이 기존의 방법에서는 트리의 깊이가 증가 할수록 훈련 데이터에 대한 오차는 줄어드는 반면, 어느 정도 깊이 이상에서는 over-fitting등으로 인하여 검증 데이터에 대한 일관성을 얻을 수 없었다. 그러나 제안된 방법에서는 노드 상호간의 정보를 고려함으로써 기존의 방법에 비하여 향상된 일관성 있는 결과를 얻을 수 있음을 보여주고 있다. 표 2에서는 모델트리의 깊이를 1에서 6까지 증가시키면서 검증데이터의 성능 변화를 나타내었다.

Abalone 데이터의 경우 RMSE(Root Mean Squared Error)를 고려할 때 기존의 방법의 경우 Std(Standard deviation)가 ±0.0042인 반면 제안된 방법에서는 ±0.0010으로 기존의 방법에 비하여 우수한 성능을 가지고 있음을 확인할 수 있으며, 다른 벤치마크 데이터들에서도 향상된 결과를 얻었다. 마지막으로 표 3에서는 기존의 방법과 제안된 방법에서의 최종 성능을 상호 비교하여 나타냈다.

표 2. 벤치마크 데이터에 대한 실험 결과(Mean)

Methods	Correlation coefficient		Root mean squared error	
	C-FMT	Proposed method	C-FMT	Proposed method
Abalone	0.6966 ±0.0251	0.7227 ±0.0065	0.0822 ±0.0042	0.0780 ±0.0010
Delta ailerons	0.8341 ±0.0049	0.8389 ±0.0024	0.0404 ±0.0005	0.0400 ±0.0003
Delta elevators	0.7963 ±0.0033	0.7988 ±0.0002	0.0531 ±0.0004	0.0529 ±0.0001
Computer activity	0.9486 ±0.0014	0.9500 ±0.0013	0.0435 ±0.0006	0.0429 ±0.0005

표 3. 벤치마크 데이터에 대한 실험 결과(Min)

Methods	Correlation coefficient		Root mean squared error	
	C-FMT	Proposed method	C-FMT	Proposed method
Abalone	0.6725	0.7178	0.0765	0.0761
Delta ailerons	0.8266	0.8287	0.0399	0.0397
Delta elevators	0.7904	0.7966	0.0528	0.0528
Computer activity	0.9462	0.9462	0.0428	0.0425

4. 결 론

본 논문에서는 기존의 클러스터 기반 퍼지 모델트리에서 트리의 깊이에 따른 훈련 및 검증데이터의 일관성 문제점을 해결하기 위해 상호 노드간의 정보를 고려하는 방법을 제안하였다. 제안된 방법은 먼저 입력과 출력변수의 속성을 고려한 퍼지 클러스터링에 의해 중심벡터를 계산한 후, 중심벡터들과 입력 속성간의 소속도를 이용하여 구간 분할된 영역별로 각각의 선형모델을 구축한다. 예측 단계에서는 입력된 데이터가 잎노드에 도달하는 노드간의 중심벡터와 입력 데이터간의 거리값에 따른 소속도를 계산한 후 최종적으로 무게 중심법을 이용하여 출력값을 예측하게 된다. 제안된 방법의 우수성을 보이기 위해 다양한 벤치마크 데이터를 대상을 실험한 결과, 기존의 클러스터 기반 퍼지 모델트리보다 향상된 성능을 보임을 확인하였다.

본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음.[2007-S-020-01, 프라이버시 보호형 바이오인식 시스템 개발]

참 고 문 헌

[1] Donato Malerbe, and et al, "Stepwise Indcution of Model Trees, LNAI 1275, pp.20-32, 2001.
[2] Quinlan J.R. "Learning with continuous classes" in Proceedings AI'92, Adams & Sterling (Eds.), World Scientific, pp. 343-348, 1992.
[3] Karalic A, "Linear regression in regression tree leaves, in Proceedings of ISSEK'92, Bled, Solvenia, 1992.
[4] Wang Y., Witten I.H., "Inducing Model Trees for Continuous Classes", in Poster Paper of the 9th European Conference on Machine Learning (ECML 97)., M. van Someren, & G. Widmer (Eds.), Prague, Czech Republic, pp. 128-137, 1997.
[5] Lanubile A., Malerba D, "Induction of regression trees with Regtree", in Book of Short Paper on Classification and Data Analysis", Pescara, Italy, pp. 253-260, 1997.
[6] Torgo L, "Kernel Regression Trees", in Poster paper of 9th European Conference on Machine Learning (ECML 97), M. van Someren, & G. Widmer (Eds.), Prague, Czech Republic, pp. 118-127, 1997.
[7] Witold Pedrycz, "C-Fuzzy Decision Trees", IEEE Trans. on System, Man, and Cybernetics, Part C, Vol. 35, No. 4, pp. 498-511, 2005.
[8] 이대종, 박진일, 전명근외, "클러스터 기반 퍼지 모델트리를 이용한 데이터 모델링", 한국퍼지 및 지능시스템학회 논문지, Vol. 16, No. 5, pp. 608-615, 2006.