

소프트 컴퓨팅 기법을 이용한 개인화된 손동작 인식 시스템

A Personalized Hand Gesture Recognition System using Soft Computing Technique

전문진, 도준형, 이상완, 박광현, 변증남

대전시 유성구 한국과학기술원 전자전산학과

E-mail: {moonjin, jhdo, bigbean, akaii}@ctrsys.kaist.ac.kr, zbien@ee.kaist.ac.kr

요 약

최근 하지가 불편한 노약자나 장애인이 집 안의 다양한 가전기기를 손쉽게 제어할 수 있게 하는 비전 기반의 손동작 인식 기술이 발전해 왔다. 다수의 사용자가 하나의 손동작 인식 시스템을 사용할 경우 사용자마다 손동작 특성이 모두 다르기 때문에 특정 사용자의 인식률이 저하되는 문제가 발생한다. 또한 동일한 사용자라 하더라도 시간에 따라 손동작 특성이 변화할 수 있다. 사용자마다 다른 손동작 특성은 모델 학습 및 선택 기법을 사용해 효과적으로 다루어질 수 있다. 시간에 따라 변하는 사용자의 특성은 퍼지 개념을 이용해 효과적으로 다루어질 수 있다. 본 논문에서는 다변량 퍼지 의사결정트리를 이용해 사용자 별 인식모델을 만드는 방법을 제시한다. 또한 새로운 사용자가 시스템을 사용할 경우 가장 적합한 모델을 선택해 인식에 사용하고 인식률을 측정한다.

Key Words : Fuzzy decision tree, Model selection, Hand gesture recognition

1. 서 론

장애인이거나 노약자를 위한 서비스의 수요가 증가하면서 보조공학 기술도 발전하고 있다. 한국과학기술원 인간친화복지로봇 연구센터에서는 장애인이거나 노약자가 집안의 다양한 가전기기를 간단한 손동작으로 제어할 수 있도록 하는 소프트 리모컨 시스템을 개발해왔다[1]. 이 시스템의 성능을 결정하는 중요한 기술 중에 하나로 손동작 인식 알고리즘을 들 수 있다. 같은 손동작 인식 알고리즘을 여러 사용자가 사용할 경우 그림 1과 같이 사용자마다 손동작 특성이 모두 다르기 때문에 특정 사용자의 손동작 인식이 저하되는 문제가 발생한다.

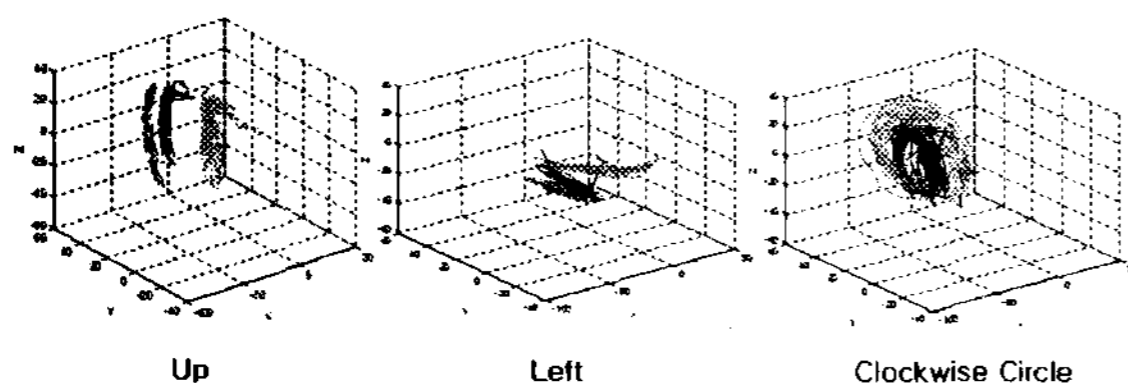


그림 1. 3명의 사용자의 손동작 특성

또한 동일한 사용자라 하더라도 시간에 따라 손동작 특성이 변화할 수 있기 때문에 고정된 결정 경계를 갖는 인식 알고리즘의 인식이 저하되는 경우가 발생하게 된다.

인식률 향상을 위해 결정 경계를 늘리면 의도된 손동작에 대한 인식률은 향상되지만 의도되지 않은 동작을 명령으로 인식해 시스템을 오동작 시키는 경우가 자주 발생하게 된다. 이러한 오동작은 false positive error이며 손동작 인식 알고리즘을 설계할 때 반드시 고려해야 하는 사항이다.

동작 인식을 위한 알고리즘으로 HMM (Hidden Markov Model)이 널리 사용되어 왔다. 수화 인식과 같은 복잡한 동작을 인식하는 경우 HMM이 가장 적합한 인식 알고리즘이라 할 수 있다[2]. 그러나 HMM은 대체로 많은 학습 데이터를 필요로 하고 다른 알고리즘에 비해 분류 능력이 뛰어나지 않다고 알려져 있다[3].

빠른 학습 시간과 뛰어난 인식률의 장점을 갖는 의사결정트리는 고정된 결정 경계 때문에 손동작과 같은 특성 변화가 심한 데이터에 대해 인식이 저하되는 단점을 갖는다. 상위 노

드에서 오분류된 데이터는 하위 노드의 속성 값이 아무리 정확해도 하위노드에서 분류 결과를 회복할 수 없다. 이러한 점은 학습데이터와 시험데이터의 특성변화가 있는, 즉 손동작과 같이 시간에 따라 변화할 수 있는 데이터일수록 더 심해진다고 할 수 있다. 이러한 단점은 의사결정트리에 퍼지 논리를 적용함으로써 해결될 수 있다.

본 논문에서는 퍼지 의사결정 트리[4,5]의 특성상 복잡해지는 노드 구조를 간소화하기 위해 다변량 퍼지 의사결정트리(Multivariate Fuzzy Decision Tree, 이하 MFDT)를 제안하고 MFDT를 이용해 각 사용자의 손동작 인식 모델을 생성하는 방법을 제시한다. 또한 여러 사용자가 시스템을 사용할 때 모델 선택을 통해 사용자에게 가장 적합한 인식 모델을 선택하고 인식률을 측정한다.

본 논문의 2장에서는 MFDT의 학습 및 분류방법을 다룬다. 3장에서는 벤치마크 데이터와 손동작 데이터를 이용해 인식률을 측정한다. 4장에서는 결론 및 추후과제에 대해 고찰한다.

2. 다변량 퍼지 의사결정 트리

일반적인 퍼지 의사결정 트리가 각 노드에 하나의 속성을 사용하는 반면 MFDT는 속성 벡터를 사용한다. 그림 2에는 단일 속성을 사용한 분기와 속성 벡터를 사용한 분기가 비교되어 있다.

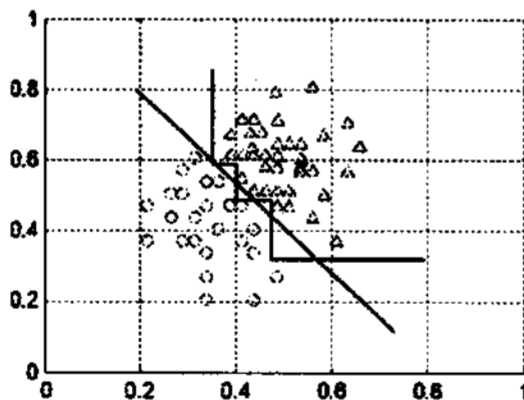


그림 2. 단일 속성 분기와 다변량 속성 분기

단일 속성을 사용하면 6회에 걸쳐 노드를 분기해야 하는 반면 속성 벡터를 사용하면 한번의 분기만으로 충분하다. 이와 같은 개념을 퍼지 의사결정트리에 적용할 수 있으며 속성 벡터는 LDA(Linear discriminant analysis)를 이용해 구할 수 있다[6].

2.1 다변량 퍼지 의사결정 트리 학습

MFDT를 학습하는 방법은 ID3 알고리즘이 사용되는 방법과 유사한 방법을 따른다. 학습 데이터가 주어지면 정보 이득을 최대화하는 관점에서 각 노드의 속성 벡터와 소속 함수를 결정한다. MFDT는 하나의 근 노드와 다수의 일반 노드, 단말 노드가 연결된 형태를 갖는다.

학습 방법은 다음과 같다.

- Step 1. 근 노드 생성
- Step 2. 정보 이득을 최대화하는 노드 생성
 - 1) LDA를 이용한 속성 벡터 w 결정

$$\text{Maximize } J(w) = \frac{w^T S_B w}{w^T S_W w}$$

$$S_B = \sum_{i=1}^K (m_i - m)(m_i - m)^T$$

$$S_W = \sum_{i=1}^K S_i$$

$$S_i = \sum_{x \in \text{class } i} (x - m_i)(x - m_i)^T$$

$$m = \frac{1}{K} \sum_{i=1}^K m_i$$

m_i : i 번째 클래스인 데이터의 평균

$\therefore w = S_W^{-1} S_B$ 의 가장 큰 고유 벡터

- 2) w 를 이용해 속성 값 계산

$$z = w^T x$$

- 3) 현재 노드의 엔트로피 계산

$$\text{Entropy} = - \sum_i P_i^{S^k} \log_2 P_i^{S^k}$$

$$P_i^{S^k} = \frac{N_{S^k}^i}{N_{S^k}}, N_{S^k} = \sum_i N_{S^k}^i$$

S^k : 현재 노드의 데이터 x 의 집합

$N_{S^k}^i$: i 번째 클래스인 데이터의 개수

- 4) 현재 노드가 w 및 가장 정보 이득이 큰 소속 함수로 분기할 경우의 엔트로피를 계산

$$S_{v|w}^i = \left\{ \begin{array}{l} (z, \mu_{S_{v|w}^k}(z)) | \mu_{S_{v|w}^k}(z) : w \text{ 위의} \\ v \text{ 번째 소속함수의 } \mu(z) \end{array} \right\}$$

$$C_{S_{v|w}^k}^i = \sum_{\substack{\text{class of } z = i \\ z \in \text{Supp}(S_{v|w}^k)}} \mu_{S_{v|w}^k}(z), C_{S_{v|w}^k} = \sum_i C_{S_{v|w}^k}^i$$

$$P_i^{S_{v|w}^k} = \frac{C_{S_{v|w}^k}^i}{C_{S_{v|w}^k}}$$

$$\text{Entropy}(S_{v|w}^k) = - \sum_i P_i^{S_{v|w}^k} \log_2 P_i^{S_{v|w}^k}$$

- 5) 현재 노드가 w 를 이용해 분기될 때의 정보 이득을 계산

$$\text{Gain}(S^k) = \text{Entropy}(S^k) - \sum_v \frac{N_{S_{v|w}^k}}{N_{S^k}} \text{Entropy}(S_{v|w}^k)$$

- 6) 한 가지 속성만 사용할 경우(univariate case)의 최대 정보이득을 갖게 하는 속성 및 소속 함수를 구함

- 7) 하나의 속성과 다변량 중에 정보이득이 큰 경우를 선택하고 선택된 속성 벡터와 소속 함수를 사용해 자식 노드를 생성

Step 3. 단말 조건을 만족하면 현재의 노드를 단말 노드로 만들고 클래스를 부과. 만족하지 않으면 모든 자식노드에서 Step 2를 재귀적으로 반복

LDA를 이용하면 그림 3와 같은 속성 벡터 w 를 구할 수 있다.

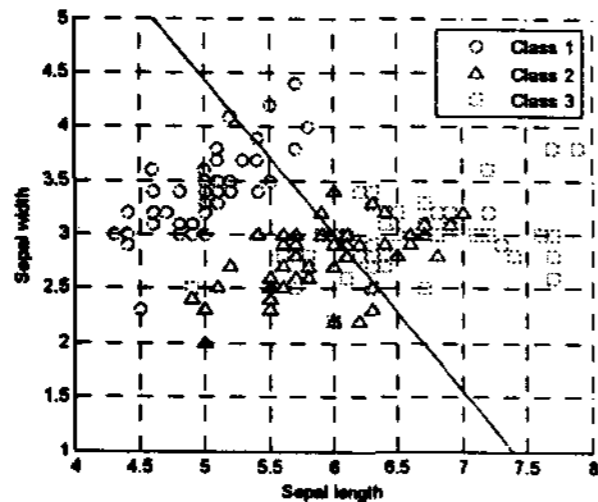


그림 3. LDA를 이용해 구한 속성 벡터 w

정보 이득을 최대화하는 퍼지 소속 함수 생성 방법을 이용해 그림 4와 같은 퍼지 소속 함수를 얻을 수 있다. 퍼지 소속 함수는 클래스의 변화가 있는 임의의 지점의 조합 중에 해당 조합의 값으로 퍼지 소속 함수를 만들고 데이터를 나눌 경우 가장 정보 이득이 큰 조합을 선택하는 방법에 의해 결정된다.

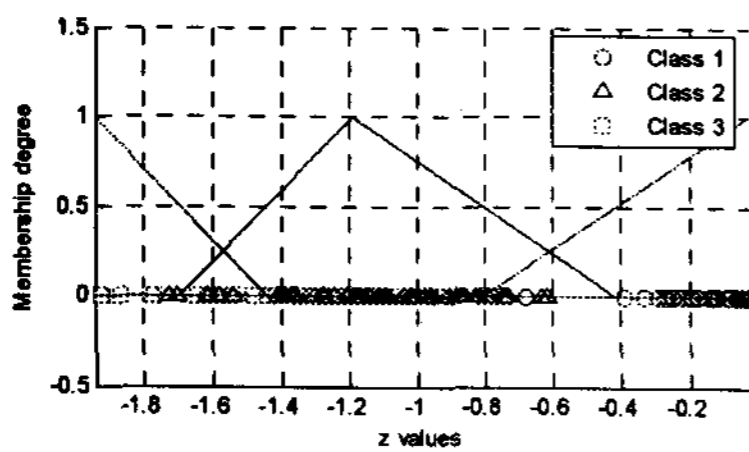


그림 4. 자동으로 생성된 퍼지 소속 함수

학습 과정을 통해 그림 5과 같은 MFDT 모델이 생성된다.

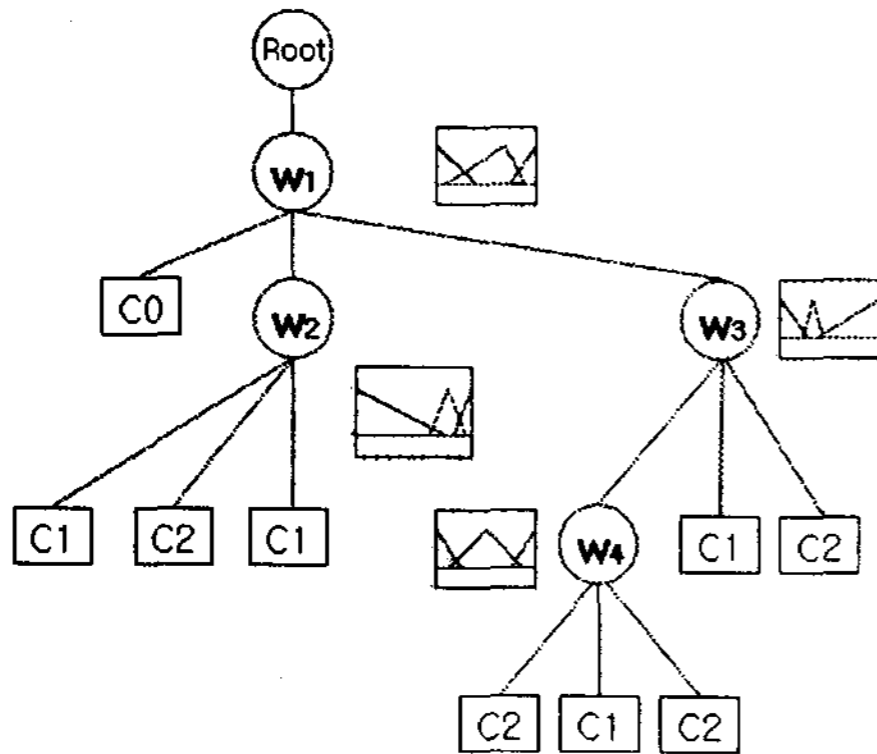


그림 5. 학습된 MFDT

2.2 다변량 퍼지 의사결정트리 분류

생성된 MFDT 모델을 이용해 새로 입력된 데이터를 분류하는 방법은 다음과 같다.

근 노드에서 각 단말노드 까지 거쳐 가는 노

드 소속 함수와 속성 벡터를 이용해 T-norm을 구한다.

$$A_n = \prod_{i=\text{root node}}^{n_{th} \text{ leaf node}} \mu_{s_{i/w}}^i(x)$$

가장 큰 T-norm을 갖는 단말노드의 클래스로 입력데이터를 분류한다.

$$\text{class } N = \underset{n}{\operatorname{argmax}} A_n$$

3. 실험 및 결과

3.1 벤치마크 데이터

제안된 MFDT 학습 및 분류 알고리즘의 일반적 성능을 측정하기 위해 벤치마크 데이터인 UCI Machine Learning Repository[7]의 Iris data와 Wine data를 사용했다. 데이터의 특징과 MFDT를 이용한 인식률을 표 1과 표 2에 나타내었다. 인식률은 5x2 fold cross validation의 평균 인식률이다.

표 1. 벤치마크 데이터

종류	클래스 수	샘플 개수	속성 개수
Iris	3	150	4
Wine	3	178	13

표 2. 벤치마크 데이터의 인식률

종류	C4.5	C5.0	FDT	MFDT
Iris	92.9%	92.9%	93.7%	96.5%
Wine	86.6%	89.2%	88.0%	89.9%

표 2를 통해 MFDT의 학습 및 분류가 벤치마크 데이터에 높은 성능을 가진다는 것을 알 수 있다.

3.2 손동작 데이터

손동작 데이터는 표 3에 나타나있는 10가지 동작을 사용했다. 총 10명의 사용자로부터 50 set의 데이터를 수집했고 각 set은 10가지 손동작으로 구성된다.

표 3. 손동작 클래스 및 속성

10가지 손동작	16가지 속성
1. 위	<ul style="list-style-type: none"> · x, y, z 축으로의 길이 · x, y, z 축에서의 최소치 · x, y, z 축에서의 최대치 · x, y, z 축에서 최소치일 때의 시간 · x, y, z 축에서 최대치일 때의 시간 · 이심률
2. 아래	
3. 왼쪽	
4. 오른쪽	
5. 앞	
6. 뒤	
7. 시계방향 원	
8. 반시계방향 원	
9. 시계방향 반원	
10. 반시계방향 반원	

손동작 인식 시스템의 성능을 나타내는 중요한 척도 중에 하나가 의도되지 않은 동작 (garbage 데이터)를 차단하는 능력이기 때문에 MFDT 학습에 10가지 손동작뿐만 아니라 다양한 일상 동작도 사용했다. 표 4에는 학습에 사용한 일상 동작을 나타내었다.

표 4. 학습에 사용된 일상 동작

<ul style="list-style-type: none"> · 물을 마시는 동작 · 책을 읽는 동작 · 신문을 읽는 동작 · 과자를 집어 먹는 동작 · 기지개를 켜는 동작
--

각 사용자 당 10 set의 손동작 데이터와 2 set의 일상 동작을 학습에 사용하여 10 개의 MFDT 모델을 생성했다. 사용자가 시스템을 사용할 때 1set의 손동작 데이터를 이용해 인식률 기반의 모델 선택을 했고 선택된 모델을 이용해 나머지 39 set을 분류했다. 인식률은 표 5와 같다.

표 5. MFDT를 이용한 모델 선택 및 인식률

결과	선택된 모델	인식률	False positive rejection rate
사용자 1	1	96.6%	82.5%
사용자 2	2	94.1%	86.3%
사용자 3	3	94.1%	55.0%
사용자 4	4	93.1%	91.3%
사용자 5	5	96.9%	93.8%
사용자 6	6	93.8%	83.8%
사용자 7	7	95.5%	90.0%
사용자 8	8	86.2%	80.0%
사용자 9	9	94.1%	92.5%
사용자 10	5	89.3%	93.8%
평균		93.4%	83.9%

결정 경계를 넓히면 인식률은 높아지지만 False positive error 역시 증가하게 되는 기존 heuristic tree의 단점을 MFDT를 이용해 극복했다. 그림 6을 통해 heuristic tree와 MFDT의 인식률 차이를 보였다.

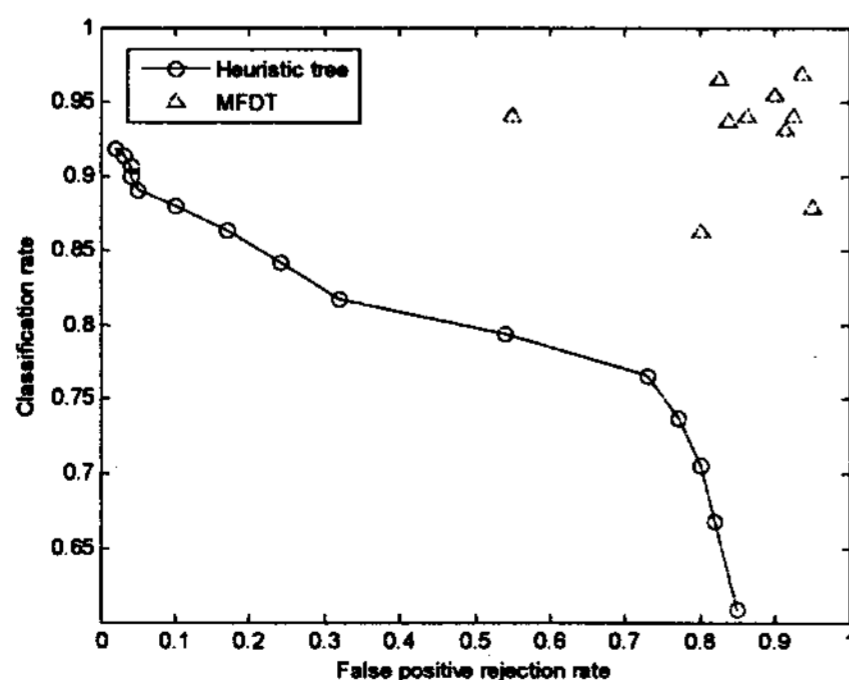


그림 6. 인식률과 false positive rejection rate의 관계

4. 결론 및 추후과제

본 논문에서는 MFDT 학습 및 분류 알고리즘을 제시했고 벤치마크 데이터 및 손동작 데이터에 적용했다. 사람마다 손동작 데이터의 특징이 다양하다는 점과 같은 사람이라도 때때로 변화하는 손동작 특성을 갖는다는 점은 퍼지개념을 이용해 효과적으로 다루어질 수 있다. 사용자가 시스템을 사용할 때 모델 선택 방법에 의해 적합한 인식모델을 선택해 높은 인식률을 얻을 수 있었다.

새로운 사용자가 등장할 경우 비슷한 모델이 없을 때 모델 선택 방법만으로는 높은 인식률을 얻을 수 없다. 이러한 점을 극복하기 위해 사용자 적용 알고리즘을 연구하고 있다. 모델 선택 후 사용자 적용 과정을 통해 인식 알고리즘을 좀 더 강인하게 설계할 수 있을 것이다.

참 고 문 헌

[1] Jun-Hyeong Do, Hyoyoung Jang, Sung Hoon Jung, Jinwoo Jung, Zeungnam Bien, "Soft Remote Control System in the Intelligent Sweet Home", *Proc. of IEEE Int. conf. on IROS*, pp. 3984-3989, 2005

[2] Sung Hoon Jung, "Incremental User Adaptation in Korean Sign Language Recognition Using Motion Similarity and Prediction from Adaptation History", Master's Thesis, KAIST, 2007.

[3] Herve Bourlard, "Links Between Markov Models and Multilayer Perceptrons", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No.12, pp. 1167-1178, Dec. 1990.

[4] Cezary Z. Janikow, "Fuzzy Decision Trees : Issues and Methods", *IEEE Transactions on systems, man, and cybernetics*, Vol. 28, No. 1, pp. 1-14, Feb 1998

[5] Koen-Myung Lee, Kyung-Mi Lee, Jee-Hyong Lee, Hyung Lee-Kwang, "A Fuzzy Decision Tree Induction Method for Fuzzy Data", *1999 IEEE International Fuzzy Systems Conference Proceedings*, Vol. 1, pp. 16-21, August 22-25, 1999, Seoul, Korea

[6] Olcay Taner Yildiz and Ethem Alpaydin, "Linear Discriminant Trees", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 19, No. 3 pp. 323-353, 2005

[7] UCI Machine Learning Repository