자기 구성 지도와 은닉 마르코프 모델을 이용한 가속도 센서 기반 행동 인식

황금성, 조성배

Activity Recognition based on Accelerometer using Self Organizing Maps and Hidden Markov Model

Keum-Sung Hwang, Sung-Bae Cho

요약 ˇ ˇ 최근 동작 및 행동 인식에 대한 연구가 활발하다. 특히, 센서가 소형화되고 저렴해지면서 그 활용을 위한 관심이 증가하고 있다. 기존의 많은 행동 인식 연구에서 사용되어 온 정적 분류 기술 기반 동작 인식 방법은 연속적인 데이터 분류 기술에 비해 유연성 및 활용성이 부족할 수 있다. 본 논문에서는 연속적인 데이터의 패턴 분류 및 인식에 효과적인 확률적 추론 기법인 은닉 마르코프 모델(Hidden Markov Model)과 사전 지식 없이도 자동 학습이 가능하며 의미 깊은 궤적 패턴을 클러스터링하고 효과적인 양자화가 가능한 자기구성지도(Self Organizing Map)를 이용한 동작 인식 기술을 소개한다. 또한, 그 유용성을 입증하기 위해 실제 가속도 센서를 이용하여 다양한 동작에 대한 데이터를 수집하고 분류 성능을 분석 및 평가한다. 실험에서는 실제 가속도 센서를 통해 수집된 숫자를 그리는 동작의 성능 평가 결과를 보이고, 행동 인식기 별 성능과 전체 인식기별 성능을 비교한다.

핵심어: Human Activity Recognition, Hidden Markov Model, Self Organizing Map, Accelerometer

1. 소개

최근 영상 및 일반적인 센서 정보 기반 사용자의 동작 및 행동 인식 연구는 활발한 편이다. 동작 및 행동 인식에 대한 연구는 비전 컴퓨팅 분야에서는 영상으로 인식된 사물 및 사람의 행동 분석을 위해서, HCI(Human and Computer Interaction) 분야 및 지능 로봇 분야에서는 사용자의 입출력 정보를 분석하기 위해서 연구가 진행되고 있다. 특히, 유비쿼터스 컴퓨팅에서 외부환경을 이해하기 위해 센서를 통한 정보 수집이 필요하다. 이때 사용자와의 상호작용이나 사용자의 명령, 상태를 이해하기 위해서는 동작 센서정보를 이해하기 위한 방법이 필요하다.

가속도 센서 기반 동작 인식을 위해 다양한 연구가 수행되어 왔는데, 기존에 수행된 동작 인식 연구는 대부분 동작 센서 데이터를 전처리한 뒤 분류 문제를 해결하는 방법이었다. 예를 들어, MIT에서는 이러한 동작 인식 문제를 풀기 위해 2004년 연구에서 가속도 센서 데이터를 전처리하여 다양한 특징 데이터(평균값, 에너지, frequency-domain entropy, correlation 등)를 추출한 뒤 결정트리(Decision Tree)를 이용하여 인간의 행동을 분류하는 방법을 사용하였다[1]. 하지만, 이러한 방법은 연속적인 동작 패턴을 분류하는 문제를 정적인 문제로 바꾸어서 해결하기 때문에 연속 데이터 인식 문제에 대한 근본적인 해결책이 될 수 없다. 연속 패턴은 그 변화의 크기와 시간이 매우 가변적이기 때문에 고정적인 특징 데이터의 분류방법으로 풀기에는 한계가 있다. 따라서 본 논문에서는 연속 동작 패턴을 고려한 동작 인식 방법을 제안한다.

본 논문에서는 연속 동작 패턴 인식을 위해 은닉 마르코프 모델(HMM; Hidden Markov Model)을 사용한다[2]. HMM은 확률기반의 추론 모델로서 시간적이고 연속적인 정보에서 패턴을 인식하고 동작 및 행동을 추론하는데 적합하다. 그리고 다양한 공간적 변화 패턴을 다룰 수 있기 때문에 동작 센서 인식에 적합한 방법이다. 이때, 동작/행동 인식 문제에 적합한 HMM 구조를 전문가가 직접 설계해 주는 것은 쉽지 않은 일이며, 적합한 설계 전략 및 학습 방법이 요구된다. 본 논문에서는 환경으로부터 주어진 센서데이터를 이용해 HMM 구조를 자동으로 학습하고 주어진 모델로부터 사용자의 동작 및 행동을 추론하는 알고리즘을 개발한다.

또한, 사전 지식 없이도 자동 학습이 가능하며, 의미 깊은 궤적 패턴을 클러스터링한 뒤 효과적인 양자화(Quantization)가 가능한 자기구성지도(Self Organizing Map [3])를 이용하여 동작 센서 데이터의 전처리한 뒤 동작 인

본 연구는 2007년 한국전자통신연구원의 연구비 지원에 의한 것임.

*주저자 : 연세대학교 컴퓨터과학과 박사과정; e-mail: yellowg@cs.yonsei.ac.kr
**교신저자 : 연새대학교 컴퓨터과학과 교수; e-mail: sbcho@cs.yonsei.ac.kr

식을 위해 활용하는 방법도 소개한다.

SOM을 이용해 궤적 데이터를 양자화한 뒤에 Markov Model을 이용하여 궤적의 종류를 인식하는 방법은 본 연구실의 이전 연구[4]에서 수행한 바 있다. 본 논문에서는 이전 연구의 연구를 확장하여 동작 센서 활용 및 HMM 기반 동작 인식을 위한 구를 수행한다.

2. SOM과 HMM 기반 연속 동작 인식

본 논문에서 제안하는 동작인식 기술은 그림 1과 같다. HMM은 시간적으로 연속적이고 지속적인 공간 변화를 포함하고 있는 연속 데이터의 분류 문제에서 좋은 성능을 발휘한다. 특히, 시작과 끝이 불분명한 동작 센서 데이터의 경우 실시간 분류가 가능한 HMM 모델의 사용이 유리하다. 또한, HMM 모델은 수학적 배경과 효과적인 학습 알고리즘이 알려져 있고, 가시적인 그래프 모델을 가지고 있기 때문에 활용의 이점이 크다. 따라서 본 논문에서는 가속도 센서에서 얻어진 정보를 통해서 연속적인 동작을 인식하기 위해 HMM 모델을 사용한다.

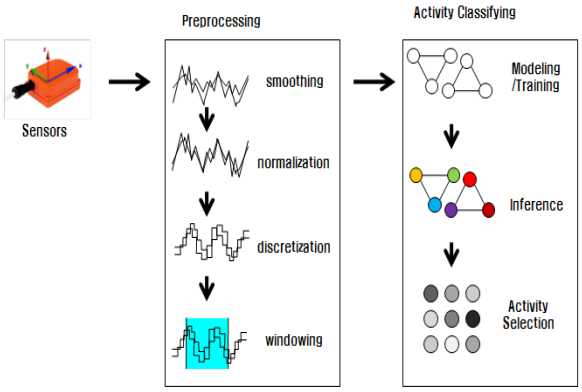

그림 1. 행동 인식기 전체 구조

이때, 연속적인 동작의 인식을 위해 사용되는 HMM은 이산화(discretization) 혹은 양자화(Quantization)된 데이터를 입력으로 사용하다. 가속도 센서에서의 방향 및 궤적 정보를 단순히 일정 구간으로 나누어 이산화할 수도 있으나, 본 논문에서는 그림 2와 같이 시공간적 특성을 잘 고려하여 반영할 수 있도록 하기 위해서 SOM을 이용한다. SOM은 신경망(Neural Network)의 일종으로서 사전 지식 없이도 자동 학습이 가능하며 의미 깊은 궤적 패턴을 찾고, 찾아진 패턴의 위상(topology)을 통해 효과적인 양자화가 가능한 방법이다[3]. SOM은 위상 유지 능력과 입력 공간 추상화 능력이 좋기 때문에 중요한 궤적 정보를 추출하고 이를 이용한 연속적인 동작 패턴 인식에 활용이 가능하다.

2.1 전처리 기술의 적용

센서를 통해 수집한 데이터의 잡음 처리를 위해서 데이터 평활화 기법을 적용하였다. 데이터 평활화의 알고리즘 중에서 사각형 평활화를 활용하였다. 실시간으로 수집되는 데이터에 대해서 일정 크기의 윈도우가 이동을 하면서 해당하는 윈도우 안에서 평활화 알고리즘을 적용하는 방식으로 진행된다. 그림 3과 같이 윈도우(w)는 현재 입력되고 있는 데이터가 계속 추가 되며 시간 t의 윈도우와 시간 t-1의 윈도우는 일정한 데이터가 겹쳐지게(overlap)하게 되어서 데이터 변동이 심화되지 않도록 해준다.

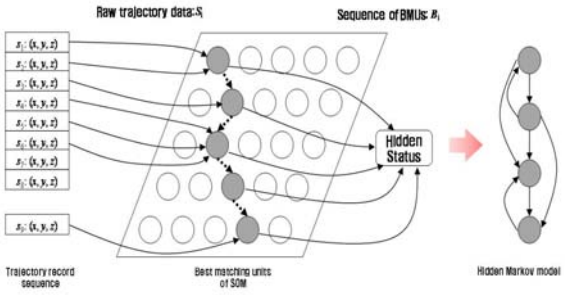

그림 2. SOM과 HMM을 이용한 동작 인식기 구조

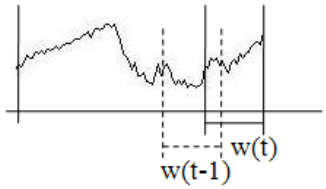

그림 3. 센서 데이터의 평활화를 위한 윈도우 정의

사각 평활화는 해당하는 윈도우 안에 있는 데이터의 평균값을 계산하는 방법으로 아래와 같은 수식으로 계산해낼 수 있다. 윈도우 안에 있는 모든 값의 합에 윈도우의 크기(s)만큼 나누면 해당하는 시간 t의 값을 결정할 수 있다.

$$f(w(t)) = \frac{\sum_k w_k}{s}, \; w_k \in w(t) \qquad (1)$$

2.2 SOM을 이용한 전처리

실시간으로 행동을 인식하기 위해서는 중요한 혹은 필요한 센서 정보만을 추출하게 되는데 추출된 다변량 데이터는 통계적인 방법에 의해 사용되어질 수 있지만 HMM과 같은 고정길이의 시퀀스를 필요로 하는 방법에는 적용하기 어렵다는 문제가 있다. 즉, 따라서 HMM과 같은 패턴인식 방법으로 행동을 모델링하기 위해서는 추출된 다차원 정보를 저차원 정보로 변환할 필요가 있는데, 이는 다차원 정보는 통계적 기법에서는 이용이 가능하지만 고정크기의 저차원 정보를 요구하는 방법에는 사용될 수 없기 때문이다.

다차원 정보를 일차원 정보로 축약 및 이산화하기 위해서는 일반적으로 통계적인 방법이 사용된다. 본 과제에서는 통계적인 기법에 의한 이산화 방법과 입력패턴에 따라 자기

조직화하여 이차원상의 대표값으로 출력해주는 SOM (Self-Organizing Map)[3]을 이용한 방법을 사용하였다. 비교사학습(unsupervised learning) 신경망인 SOM은 다차원 입력벡터를 Euclidean distance와 같은 유사도 측정을 통해 자기조직화하고 입력값에 가장 가까운 대표값으로 출력해준다.

SOM/HMM을 이용한 행동인식기에서는 SOM을 이용해서 데이터 이산화 및 축약 방법을 사용한다. SOM을 이용하여 데이터를 축약하기 위해서는 센서로부터 추출된 정보의 크기를 정규화 하여 SOM의 입력으로 사용한다. SOM을 이용하여 이러한 입력값을 잘 분류해내기 위해서는 학습과정이 필요하다. SOM의 학습과정은 매개변수들과 가중치를 초기화하고 조건이 만족할 때까지 입력된 데이터와 유사한 것으로 대표값이 결정되도록 가중치를 갱신하는 반복과정으로 볼 수 있다. 일반적인 자기조직화 신경망의 학습 알고리즘은 다음과 같다. 수식에서 i(x)는 입력에 가장 잘 일치하는 값이며, Λi는 이웃 함수, η 는 학습률을 의미한다.

1. 가중치 벡터들(wj(0))의 초기화 (n=0).
2. 유사도 비교.

$$i(\mathbf{x}) = \arg\min_j \left\| \mathbf{x}(n) - \mathbf{w}_j \right\|$$

3. 가중치 벡터들의 갱신.

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)\Lambda_{i(\mathbf{x})}(n, j)\left(\mathbf{x}(n) - \mathbf{w}_j(n)\right)$$

4. 조건을 만족할 때까지 2 ~ 4반복

이를 도식화하면 다음 그림과 같다.

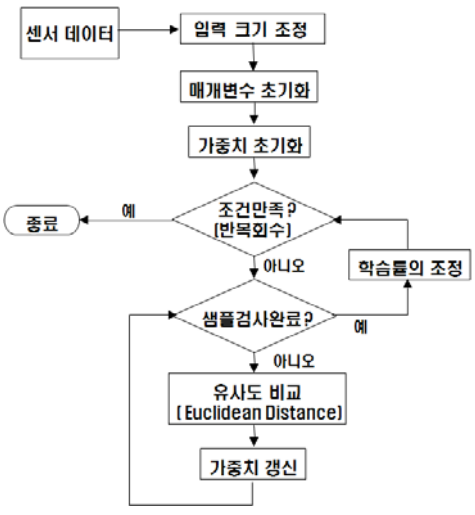

그림 4. SOM을 이용한 데이터 축약

2.3 HMM을 이용한 행동 인식기

연속데이터를 대상으로 하고 패턴인식 방법으로 HMM을 이용한 행동인식 방법을 소개한다(그림 5). 그리고 행동인식을 위한 HMM 모델링 과정을 나타낸다. 수집된 센서 데이터는 전처리를 통해 HMM 행동 인식 모듈을 통해 찾고자 하는 시퀀스 패턴을 찾는다. 이때, HMM은 행동별로 구성되어야 하며, 가장 확률값이 높은 행동이 최종 행동으로 선택되는 구조이다.

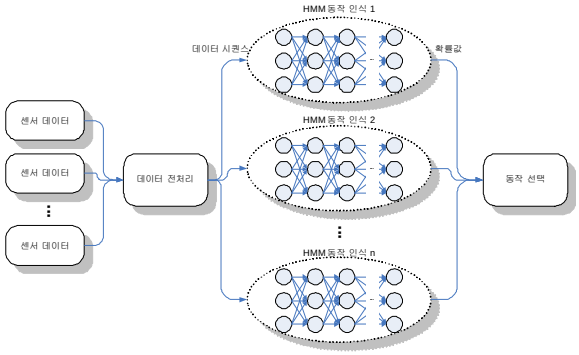

그림 5. HMM 기반 행동 인식기 구조

은닉 마르코프 모델(Hidden Markov Model; HMM)[18][19]은 유한 상태 오토마타에 확률 개념을 도입한 것으로 시간의 흐름에 따라 연속적으로 나타나는 관측치를 인식하는 통계적 기법이다. HMM은 실제적인 생성모델을 알 수 없고 단지 생성된 시퀀스에 의해서만 확률적으로 관측할 수 있는 이중으로 확률적인 절차로서, 사용자의 행위시퀀스를 모델링하기에 유용한 도구이다.

HMM은 고정된 값인 관찰 시퀀스의 길이, 상태수, 심볼수와 학습에 의해 조정되는 전이확률, 관측확률, 초기상태분포로 구성이 된다. 전이확률은 한 상태에서 다음상태로 전이할 확률을 나타내며, 관측확률은 한 상태에서 특정 심볼이 관측될 확률을 나타낸다. 초기 상태 분포는 처음에 해당 상태에서 시작할 확률을 나타낸다. HMM은 상태(state)라 불리는 노드와 노드 간의 전이를 나타내는 선분으로 구성된 그래프로 표현될 수 있다. 그래프의 각 노드는 공간적인 특성을 모델링하는 관측심볼 확률분포와 초기상태 확률분포를 가지며, 각 선분은 관측열의 시간적인 특성을 모델링하는 상태전이 확률분포를 갖는다. HMM 모델 는 학습에 의해 조정되는 변수들로 간략히 (A, B, π)로 나타낸다. 표 1은 HMM을 구성하는 파라미터를 보여준다.

HMM을 이용하여 행동을 인식하기 위해서는, 관찰된 심벌의 시퀀스 $O = O_1, O_2, \cdots, O_T$ 와 모델 $\lambda = \{A, B, \pi\}$ 가 주어졌을 때, 모델에서 대한 심벌이 나타날 확률 $P(O|\lambda)$ 를 구해야 한다. $P(O|\lambda)$ 를 계산하는 가장 확실한 방법은 모든 가능한 상태 변화 시퀀스 $Q = q_1, q_2, \cdots, q_T$ 에 대해서 $P(O|Q, \lambda)$ 를 각각 계산한 후, 이 값들을 모두 더하는 것이다.

$$P(O \mid Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \cdots b_{q_T}(O_T) \quad (2)$$

$$P(Q \mid \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \quad (3)$$

표 1. HMM 구성 파라미터

파라미터	설명
$\lambda = \{A, B, \pi\}$	HMM의 전체 모델
[redacted]: 상태번호	상태 전이 확률, aij는 상태 i에서 상태 j로 전이할 확률
$B = \{b_i(k) \mid i$: 상태 번호, k : 심볼 번호}	관측 심볼의 확률 분포, $b_i(k)$는 상태 i에서 심볼 k가 관측될 확률
$\pi = \{\pi_1, \pi_2, \cdots, \pi_N\}$	첫 번째 심볼이 각 상태에서 나타날 확률
N	상태의 개수
M	관측 심볼의 개수
T	관측된 데이터의 길이
$Q = \{q_1, q_2, \cdots, q_N\}$	내부 상태 집합
$V = \{v_1, v_2, \cdots, v_M\}$	관측 심볼 집합
O_t	시간 t에서 관측된 심볼

$P(O \mid Q, \lambda)$와 $P(Q \mid \lambda)$는 수식 (2), (3)과 같이 구할 수 있으므로, $P(O \mid \lambda)$는 수식 (4)와 같이 계산된다.

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda) P(Q \mid \lambda) \quad (4)$$

그러나 이와 같이 계산을 하려면, 하나의 덧셈을 위해서 $2T-1$번의 곱셈이 필요하며, 가능한 상태 변화 시퀀스도 N^T개만큼 존재하므로, 계산 복잡도가 $O(TN^T)$가 되어 매우 많은 연산량을 필요로 한다. 뿐만 아니라 0~1사이의 값을 여러 번 곱하게 되어, 언더플로우(underflow)를 유발할 가능성도 존재한다. 이러한 문제들을 해결하기 위한 방법으로 여기서는 Forward/Backward 알고리즘[3]을 사용한다. Forward 알고리즘은 아래 그림과 같은 흐름으로 확률적 연산을 수행하여 주어진 심볼이 나타날 확률을 계산한다.

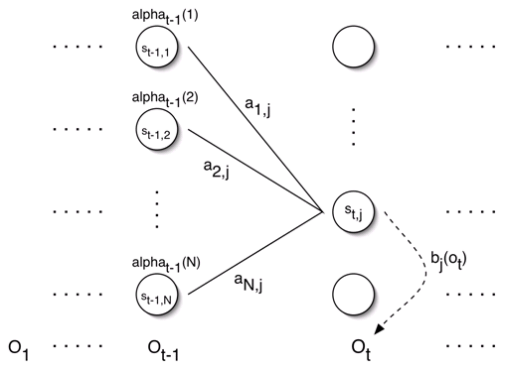

그림 6. Forward 알고리즘의 계산 흐름

이 알고리즘은 아래와 같은 루틴을 통해 계산하는 방법이다.

표 2. Forward 알고리즘 계산 흐름

- 정의: $\alpha_t(i) = P(o_1 \ldots o_t, q_t = s_i \mid \lambda)$
- 초기화단계: $\alpha_1(i) = \pi_i b_i(o_1) \quad 1 \le i \le N$
- 반복 연산 단계(종료 조건 만족시까지 반복):

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(o_t), \quad (2 \le t \le T, 1 \le j \le N)$$

- 종료 조건: $P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$

3. SOM과 HMM 기반 연속 동작 인식 실험

본 논문에서는 제안하는 방법의 유용성을 검증하기 위해 실제 가속도 센서를 이용한 실험을 수행한다. 사람의 팔에 3축 가속도 센서를 부착하고 숫자 0에서 9까지(행동0~행동9)를 인식하는 데이터를 수집하였다. 각 동작별로 50회의 데이터를 수집하였으며, 총 500개의 데이터가 수집되었다. 그리고, 데이터 중에서 40개는 학습용으로 10개는 평가용으로 사용하였다.

3.1 HMM을 이용한 행동 인식기 모델링

동작 인식을 위한 HMM을 학습하기 위해서는 수집된 센서 데이터를 통해 모델을 자동으로 구축하는 Baum-Welch Algorithm [3]이 사용된다. 이 알고리즘은 관찰된 심볼 시퀀스를 가장 잘 만족하는 HMM을 찾는 알고리즘이다. Baum-Welch Algorithm은 주어진 훈련데이터를 가장 잘 나타내는 HMM 모델을 자동으로 학습하기 위해 사용되며, 두 개의 변수가 추가로 사용된다. $\xi_t(i,j)$는 시간 t에 상태 q_i에 있다 시간 t+1에 상태 q_j에 있을 확률로 정의되며 다음과 같이 표현될 수 있다.

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\Pr(O \mid \lambda)} \quad (5)$$

$$\xi_t(i,j) = \Pr(i_t = q_i, i_{t+1} = q_j \mid O, \lambda) \quad (6)$$

$\gamma_t(i)$는 시간 t에 상태 q_i에 있을 확률이며 다음 수식을 통해 구할 수 있다.

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \quad (7)$$

두 값을 시간 t에 대해 각각 합을 취하면 하나의 시퀀스에서 각각 상태 i에서 j로 변할 기대값과 상태 i에 있을 기대값을 구할 수 있다. 위의 값이 구해지면 수식(8~10)에 의해서 새 모델 $\overline{\lambda} = (\overline{a}, \overline{b}, \overline{\pi})$를 구할 수 있다. 시퀀스 O를 관찰한 결과로 $\overline{\lambda}$를 구한 후 $\Pr(O \mid \lambda)$와 $\Pr(O \mid \overline{\lambda})$를 비교한다. $\Pr(O \mid \lambda)$가 더 크다면 우도

함수의 임계점에 다다랐으므로 재추정 과정을 종료한다. $\Pr(O|\overline{\lambda})$가 더 큰 경우는 더 나은 모델이 생성된 경우이며 λ를 $\overline{\lambda}$로 대치한 후 재추정 과정을 반복한다.

$$\overline{\pi}_i = \text{시간}(t=1)\text{에 상태 } S_i\text{인 빈도} = \gamma_1(i) \quad (8)$$

$$\overline{a}_{ij} = \frac{\text{상태 } S_i\text{에서 } S_j\text{로 전이 기대횟수}}{\text{상태 } S_i\text{에서 전이 기대횟수}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \quad (9)$$

$$\overline{b}_j(k) = \frac{\text{상태 } j\text{에서 심볼 } v_k\text{ 기대횟수}}{\text{상태 } j\text{에 있을 기대횟수}} = \frac{\sum_{t=1 s.t. O_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)} \quad (10)$$

3.2 전처리 과정에서 SOM의 맵 크기 결정 방법

SOM에서의 반복회수는 100,000으로 하였으며 학습률은 0.02로, 거리 측정은 유클리드(Euclid distance) 거리로 하여 실험하였다. 학습과 평가를 위해 사용된 데이터는 동작 센서로부터 추출된 10개 행동 500개의 시퀀스 데이터이다. 이때, SOM에서 맵 크기 결정이 중요한데 맵의 크기 결정을 위해 맵의 크기변화에 따른 양자화 에러(Quantization error)와 맵의 사용률을 실험으로 알아보았다.

그림 7에서 알 수 있듯이 SOM에서 맵의 크기가 증가할수록 양자화 에러 값이 줄어들지만 맵의 크기가 증가하게 되면 그림 8과 같이 사용되지 않는 영역이 늘어나게 되므로 사용되지 않는 영역이 작으면서 양자화 에러 값이 작은 맵을 선택하는 것이 중요하다. 양자화 에러를 계산하는 방법은 다음 수식과 같다. 여기에서 Qerr는 양자화 오류를 나타내고, k는 SOM에 들어가는 데이터의 수를 의미하며, dk는 k번째 데이터를 의미한다. 그리고 Cr()은 현재 데이터가 클러스터링 된 셀의 대표 클래스를 의미하고, d'는 데이터 d의 클래스를 의미한다. Q() 함수를 통해 현재 들어온 데이터가 클러스터링 된 셀의 대표값과 일치한지를 비교한 뒤에 이를 바탕으로 오류를 계산하는 방법이다.

$$Q_{err} = \frac{\sum_k Q(d_k)}{k}$$

$$Q(d) = \begin{cases} 0, & if\ d' = C_r(d_k) \\ 1, & f\ d' \neq C_r(d_k) \end{cases}$$

적당한 맵의 선정을 위해서 SOM의 맵의 크기를 변화시키면서 양자화 오류와 맵의 사용도를 알아보았다.

실험 결과, 맵의 크기가 9인 경우부터 양자화 에러가 10% 미만으로 줄어들고 있음을 알 수 있다. 따라서 맵의 크기는 3x3 이상이 되는 것이 좋을 것으로 보인다. 그림 8은 맵의 크기에 따라 사용되지 않은 셀의 수를 나타낸다. 맵의 크기를 1x1에서 10x10까지 변화시켰을 때, 전혀 사용되지 않은 셀의 수는 모두 0으로 나왔고, 1%이하만 사용된 경우, 2%이하만 사용된 경우, 4%이하만 사용된 경우는 그림 8과 같이 나타났다. 균등 분할인 경우 셀 당 사용률이 (1/맵의 크기)가 되는 것을 생각한다면, 전체적으로 사용률이 나쁘지 않은 것을 알 수 있다. 1% 미만 사용되는 셀이 적도록 하기 위해서는 맵의 크기가 6x6미만인 것이 좋을 것으로 보인다.

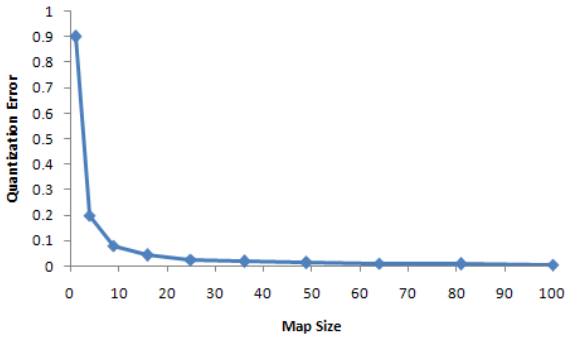

그림 7. 양자화 오류 변화

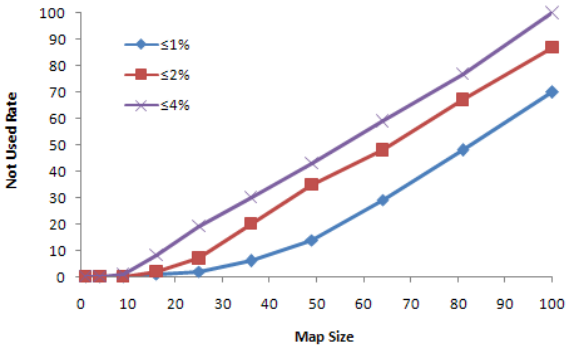

그림 8. 맵의 크기 증가에 따른 사용도 변화

실험 결과 SOM의 맵 크기는 3×3, 4×4, 5×5, 6×6 정도가 가장 이상적일 것으로 보인다.

3.3 행동별 행동인식기 성능 평가

각 행동인식기를 돌려서 얻은 최적 파라미터를 바탕으로 행동별 행동인식기 성능을 평가하였다. 평가를 위해 사용된 기준은 인식률 대비 False Positive Error 이다. 즉, 목표 인식률을 얻기 위해 필요한 False Positive Error 허용 수치를 표현하였다. 그림 9~11은 이러한 결과를 나타내고 있다. 지면관계상, 실험 결과 중에서 행동 1, 2, 3의 경우에 대해서만 그림에 표현하였다.

그림 12는 모듈별 인식 성능을 비교하기 위해 정렬하여 비교한 그래프이다. 행동 0, 6, 1, 7, 2, 9까지는 비교적 잘 인식이 되지만, 행동 3, 4, 8, 5는 False Positive Error가 큰 것을 알 수 있다. 이것은 행동 3, 4, 8, 5의 모델이 다른 행동 모델과 잘 구분이 되지 않는다는 것을 의미하는데, 실제로도 3, 4, 8, 5는 숫자를 그리는 과정이 상당히 복잡하다.

따라서 행동 시퀀스가 큰 경우에는 성능이 상대적으로 좋지 않을 수 있음을 알 수 있다.

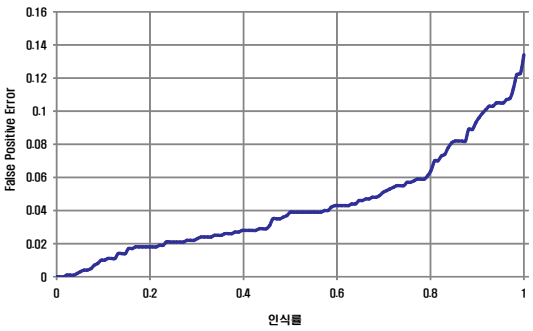

그림 9. 행동 모듈별 행동 인식 성능 평가 (행동 1, SOM 5x5, 상태수 5, 윈도우 크기 15)

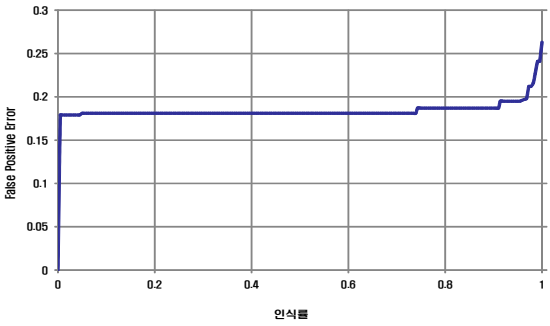

그림 10. 행동 모듈별 행동 인식 성능 평가 (행동 2, SOM 3x3, 상태수 5, 윈도우 크기 25)

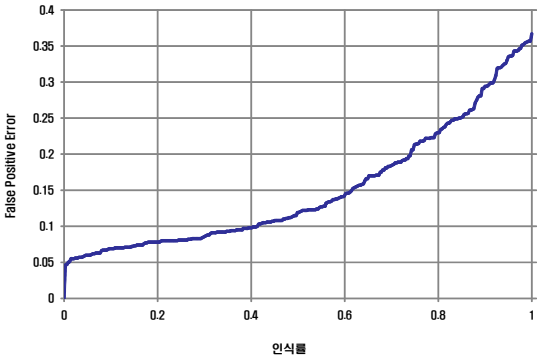

그림 11. 행동 모듈별 행동 인식 성능 평가 (행동 3, SOM 3x3, 상태수 5, 윈도우 크기 20)

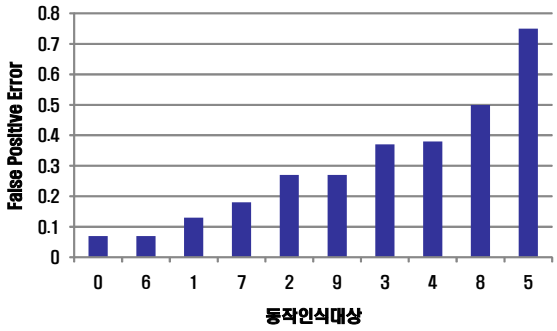

그림 12. 행동 모듈별 행동 인식 성능 평가 (종합)

3.4 전체 행동인식기 성능 비교

같은 파라미터 조건에서 생성된 행동인식기를 사용하여 행동을 구별한 경우의 성능을 분석해 보았다. 행동 인식기별로 파라미터를 최적화 한 결과가 아니기 때문에 잘못 학습된 행동 인식기가 있을 경우 좋지 않은 성능이 나올 수 있다. 표 3의 경우 일부 행동의 경우 상대적으로 좋은 성능을 보이고 있지만, 다른 행동은 거의 인식을 못하고 있다. 학습 파라미터와 윈도우 크기를 동일하게 구성한 경우 비슷한 시퀀스가 포함된 행동의 구별이 잘 이뤄지지 않기 때문이다. 따라서 여러 행동 인식기를 동시에 사용할 경우에는 이를 구분하기 위한 연구 및 행동별 파라미터 최적화가 요구된다고 볼 수 있다.

표 3. 파라미터(Som: 4x4, window 크기: 25, HMM state 수: 5인 경우)에서 학습된 행동인식기 성능 비교

	m0	m1	m2	m3	m4	m5	m6	m7	m8	m9	hit_rate
m0	15	15	8	0	120	16	8	46	3	0	0.064935
m1	4	1	5	0	3	1	5	100	17	0	0.007353
m2	3	2	110	59	30	0	13	0	5	0	0.495495
m3	9	0	7	83	26	0	64	0	0	0	0.439153
m4	25	0	117	10	8	0	4	25	10	11	0.038095
m5	0	0	4	30	0	44	24	0	171	4	0.158845
m6	0	0	0	0	0	59	0	0	130	0	0
m7	0	0	0	0	0	114	0	0	2	0	0
m8	0	0	0	0	0	1	0	0	325	0	0.996933
m9	0	0	0	0	0	8	0	0	127	0	0
											0.286526

4. 결론 및 향후 연구

본 논문은 가속도 센서를 바탕으로 인간의 연속적인 동작 패턴을 인식하기 위해 SOM과 HMM을 이용한 방법을 소개한다. 실험에서는 다양한 동작에 대해 실제로 인식 수행된 결과를 살펴본다. 제안하는 방법에서 연속 패턴을 인식하기 위해 사용된 HMM 모델은 연속 시퀀스가 길어질수록 패턴의 인식이 어려워지고 비슷한 행동 사이의 인식이 어려운 문제가 있었다. 향후에는 HMM의 계층적으로 분할하여 동작의 조합에 따른 복잡한 동작 인식 연구가 필요할 것이다.

※ 참고문헌

[1] L. Bao, S. S. Intille, "Activity recognition from user-annotated acceleration data," Lecture Notes In Computer Science, vol 3001, pp. 1-17, 2004.

[2] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. of the IEEE, Vol. 77, No. 2, 1989.

[3] T. Kohonen, "The self-organizing map," Proc. IEEE, vol. 78, no. 9, pp. 1464-1480, 1990.

[4] S.-J. Han, and S.-B. Cho, "Predicting user's movement with a combination of a self-organizing map and markov model," Lecture Notes in Computer Science, vol. 4132, pp. 884-893, 2006.