

다양한 스마트 공간에 확장 가능한 다중 센서 상태 변화

감지 기반 태스크 분할 기법

박시환[○], 김현주, 이동만

전산학부, 한국과학기술원

{psh150204, iplay93, dlee}@kaist.ac.kr

Scalable task segmentation method based on change point detection of multi-sensors in smart spaces

Sihwan Park[○], Hyunju Kim, Dongman Lee
School of Computing, KAIST

요 약

스마트 공간에서 정확한 사용자 의도 파악을 위해서는 태스크 분할이 선행되어야 한다. 이를 위해 기존의 연구들은 사람이 정의한 지식 또는 데이터 마이닝을 이용한 태스크의 패턴 기반 방법을 주로 이용하였다. 하지만 이 기법들은 다양한 스마트 공간에서 확장성 있게 적용하기 어렵다는 한계점을 가지고 있다. 이를 극복하기 위해, 다중 센서들의 상태 변화만을 감지하여 태스크 분할 여부를 판단하는 기법을 제안한다. 제안하는 기법의 성능 평가를 위해 실제 테스트베드에 기반한 다섯 가지의 센서 데이터를 이용하여 본 기법의 정확도와 실행 시간에 대해 측정하였다.

1. 서 론

스마트 공간에서 IoT 기기들이 생성하는 수많은 센서 데이터들을 이용하여 사용자 의도를 인지하기 위한 지원 기술로 센서 스트림 데이터 내의 태스크 분할 기법이 중요하다. 태스크 분할 (Task segmentation)이란 일련의 센서 스트림 데이터 시점 중에서 사용자가 행하는 태스크의 시작 시점과 종료 시점을 알아내는 것으로, 정확한 사용자 의도 패턴을 구하기 위해서는 정확한 태스크 분할이 선행되어야 한다. 이와 관련된 이전 연구들에서는 주로 태스크의 패턴을 미리 알아내고, 그에 따라 센서 스트림 데이터 중에서 태스크의 시작 패턴 혹은 종료 패턴과 비슷한 시점을 찾아내는 방법을 택하였다. 이전에 구축된 태스크의 패턴을 이용하여 태스크 분할 시점을 찾아내는 방법에는 크게 지식 기반 (knowledge driven) 분할 기법과 데이터 마이닝된 태스크 패턴 기반 (data driven) 기법이 있다.

지식 기반의 태스크 분할 기법을 사용한 [1,2]는 센서 데이터 스트림이 들어오면, 사람이 미리 구축한 태스크 온톨로지 [3]를 이용하여 계층적으로 사물 센서를 확인하고 마지막에 도달하였을 때 나타난 태스크가 있다면 태스크가 나타난 시점으로 간주하게 된다. 그러나 이 방법은 지식이 미리 정의되지 않은 새로운 공간에서는 사용이 어렵고, 미리 정의되지 않은 태스크는 스트림에서 분할될 수 없다. 또한, 이와 관련된 이전 논문들은 주로 태스크에 의존적인 사물 센서 중심으로 구성되어 있어서, 환경 센서가 같이 구축된 공간에서는 제시한 기법을 바로 적용하기에는 무리가 있다. 학습된 태스크 패턴을 기반으로 태스크 분할을 하는 [4,5]는 공간에서 축적된 데이터를 이용하여 태스크 별로 시작 시점 패턴을 추출한 후, 센서 스트림이 일치하는 시점이 있으면 그 부분을 태스크 시작 시점으로 분할한다. 그러나 이 방법은 정확한 패턴 추출을 위해 충분한 양의 안정된 데이터가 확보되어있어야 하고 데이터 수집 공간의 도메인이나 설치된 센서 구성이 바뀔 때 마다 기존에 학습된 모델을 재사용하지 못하고 데이터 수집부터 모델 학습까지 다시 수행해야한다는 한계가 있다.

이전 연구들의 한계점들을 극복하기 위해서, 본 연구에서는 Change Point Detection [6]에서 사용되는 통계적 기법을 이용하여 스마트 공간 내의 찾아낸 다중 스트림 상태 변화를 기반

태스크 분할 기법을 제안한다. 이 기법을 기반으로 본 연구에서 기여할 수 있는 부분은 다음과 같다.

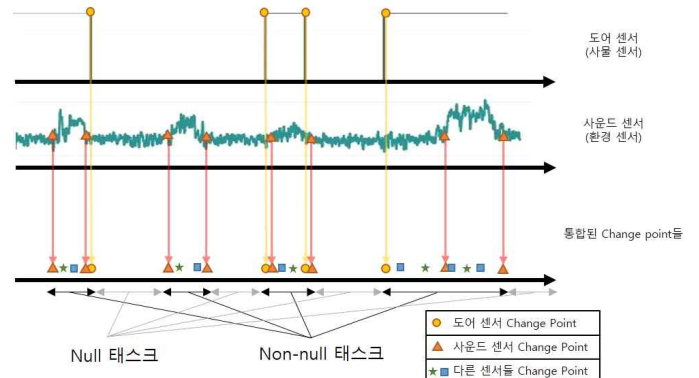


그림 1. 전체 시스템 작동 원리

1) 센서 데이터 스트림들을 통계적으로 분석하여 센서들의 상태 변화만을 기반으로 태스크 분할 여부를 검출하기 때문에 센서 데이터 외에 미리 정의된 지식이나 태스크 패턴이 필요하지 않아 센서의 종류나 공간에 구애받지 않고 적용이 가능하다.

2) 태스크 분할 여부 검출을 위해서 간단한 통계적 분석만을 요구하기 때문에 기존의 머신 러닝 기반 기법들보다 계산량이 적어 가볍다(light-weight).

2. 시스템 설계

본 연구에서의 다중 센서 데이터 기반 태스크 분할 알고리즘은 크게 다음과 같은 단계로 구성된다.

1) 각 센서 별 데이터들을 분석하여 상태 변화 시점을 찾는 Change Point Detection을 수행한다.

2) 수집된 각 센서 별 Change Point (상태 변화 시점)들을 하나의 Time Line으로 통합한다.

3) 통합된 Change Point들을 분석하여 태스크가 없는 시간

(Null Task)과 태스크가 있는 시간(Non-null Task)으로 Time Line을 분할한다.

$$d(t, w) = c(T_{t-w, t+w}) - c(T_{t-w, t}) - c(T_{t, t+w})$$

즉, 특정 시간 간격에서 비용 함수는 해당 시간 간격에 해당하는 스트림 데이터들의 표본 분산(Sample Variance)이며, 특정 시점에서 차이 함수는 해당 시점을 기준으로 전체 표본 분산과 양 쪽의 표본 분산의 합 사이의 차이를 의미한다.

3.2 다중 센서 상태 변화 기반 태스크 분할 알고리즘

본 연구에서의 다중 센서 상태 변화 기반 태스크 분할 알고리즘은 다음과 같은 전제를 갖는다.

Non-null 태스크가 진행 중인 경우, 사람들의 행동이나 상황에 따라 센서의 값이 지속적으로 변화하기 때문에 실제로 태스크가 종료되지 않았음에도 여러 Change Point들이 만들어 질 수 있다. 하지만 Null 태스크의 경우, Null 태스크가 진행 중인 시간동안 어떠한 Change Point도 발생하지 않는다. (단, 스마트 공간 외부의 환경 변화로 Change Point가 발생할 수 있지만 이러한 경우는 고려하지 않는다)

이러한 전제를 기반으로 알고리즘 2와 같이, 다음과 같은 과정을 통해 Change Point들의 패턴을 분석한다.

알고리즘 2. 다중 센서 상태 변화 기반 태스크 분할

Mltiple-Sensor-Segmentation ($D(T_{a-w, b}), UCP, T_{a, b}, w, \alpha, h$)

$I \leftarrow \text{Decision-initial-Task}(D(T_{a-w, a}), a, w, \alpha)$

$NT \leftarrow \emptyset$; Set of Null Tasks

$NNT \leftarrow \emptyset$; Set of Non-null Tasks

$S = a$; start point of task

for $i = 0$ **to** $\text{length of } UCP - 1$ **do**

if $I = \text{Null Task}$ **then**

$NT \leftarrow NT \cup \{[S, UCP[i]]\}$

$I \leftarrow \text{Non-null Task}$

$S \leftarrow UCP[i]$

else then

if $UCP[i+1] - UCP[i] > h$ **then**

$NNT \leftarrow NNT \cup \{[S, UCP[i]]\}$

$I \leftarrow \text{Null Task}$

$S \leftarrow UCP[i]$

return NT, NNT

1) 단일 센서 상태 변화 감지 알고리즘을 통해 얻은 개별 센서의 Change Point들을 하나의 Time Line으로 통합한다.

2) 시작 시점의 태스크가 Null Task인지 Non-null Task인지 판단한다. 알고리즘 3에서 볼 수 있듯이, 이는 주변 환경 센서의 값을 기준으로 한 가설 검정과 시작 시점 이전의 데이터들의 분석을 통해 이루어진다.

알고리즘 3. 시작 시점의 태스크 판별

Initial-Task-Decision ($D(T_{a-w, a}), T_{a, b}, w, \alpha, \mu_{null}$)

$CP \leftarrow \text{Change Point Detection for } D(T_{a-w, a})$

if $CP = \emptyset$ **then**

$t \leftarrow t(D(T_{a-w, a}), \mu_{null})$

if $|t| < |t_{\alpha/2, b-a}|$ **then** Null Task

else then Non-null Task

else then Non-null Task

시작 시점의 태스크 판별에 사용될 가설 검정($t - \text{Test}$)을 위해 추가적으로 다음과 같은 함수를 정의한다.

$$t(D(T_{a, b}), \mu_0) = \frac{\bar{X} - \mu_0}{s / \sqrt{b-a+1}} ; \text{test statistic}$$

$$\bar{X} = \frac{1}{b-a+1} \sum_{t=a}^b D(t), s = \sqrt{\frac{\sum_{t=a}^b (D(t) - \bar{X})^2}{b-a}}$$

구체적인 알고리즘 설계를 위해 다음을 먼저 정의하였다.

1) $T_{a, b}$: 시간 a부터 b까지의 모든 시각의 집합.

2) $D(T_{a, b})$: 시간 a부터 b까지의 스트림 데이터.

3) $D(t)$: 시각 t의 스트림 데이터.

4) $c(T_{a, b})$: 시간 간격 $[T_a, T_b]$ 의 비용 함수.

5) $d(t, w)$: 주어진 Time Window 크기 w에 대해 시점 t에서의 차이 함수.

6) w : Time window의 크기.

7) α : 가설 검정에서 적용되는 유의 수준.

8) μ_{null} : Null 태스크일 때 주변 환경 센서의 평균 값.

9) CP : Change Point들의 집합.

10) UCP : 여러 센서 스트림 데이터들로부터 검출된 Change Point들의 합집합(Unified Change Point).

11) h : Non-null 태스크에서 Null 태스크로 전환될 때, 어떠한 Change Point도 존재하지 않아야하는 시간 간격의 크기.

3. 다중 센서 데이터 기반 태스크 분할 알고리즘

3.1 단일 센서 상태 변화 감지 알고리즘

스마트 공간에서 사용되는 센서의 종류를 크게 두 가지로 분류할 수 있다. 이에 따라 단일 센서 상태 변화 감지 알고리즘은 센서의 종류에 따라 다른 알고리즘을 적용되었다.

1) 사물 센서 (Object Sensor) : 어떠한 사물에 부착, 혹은 내장되어 작동하는 센서. 주로 어떤 기기나 장비의 상태를 나타낸다 (전원 켜짐 혹은 전원 꺼짐 등). 센서 상태가 변경되었다는 것은 사용자가 센서를 사용하는 행위를 시작하거나 종료하는 것으로 바로 간주할 수 있다.

2) 주변 환경 센서 (Ambient Sensor) : 어떠한 공간의 주변 환경을 측정하는 센서. 주로 온도, 소리, 습도, 밝기 등과 같은 데이터를 측정한다. 스마트 공간에서 측정되는 센서 값이 유의미한 변화를 나타낼 경우, 해당 공간에서 새로운 행위가 일어난다는 것을 의미한다. 다음과 같은 간단한 통계적 기법을 활용하여 환경 센서의 상태 변화 시점(Change Point)을 검출할 수 있다.

본 연구에서는 알고리즘1에서 서술된 일련의 과정을 거쳐 Change Point들을 검출한다.

알고리즘 1. 주변 환경 센서 상태 변화 감지

Ambient-Change Point Detection ($D(T_{a, b}), w$)

$Z \leftarrow \emptyset$

for $t = a + w$ **to** $b - w$ **do**

$p = c(T_{t-w, t})$

$q = c(T_{t, t+w})$

$r = c(T_{t-w, t+w})$

$Z[t] \leftarrow |r - (p + q)|$

$\text{points} \leftarrow \text{Peak-Search}(Z, a, w)$

1) 전체 Time Window 크기의 1/10이하의 크기를 갖는 Time Window를 만든다.

2) 이 Time Window를 관찰 대상 시간의 처음부터 끝까지 슬라이딩 하며 각 시점의 차이 함수를 계산한다.

3) 차이 함수 값이 극댓값(Local Maxima)이 되는 시점들을 검출한다.

알고리즘1에서 사용되는 비용 함수와 차이 함수는 아래와 같이 정의하였다.

$$c(T_{a, b}) = \frac{1}{b-a} \sum_{t=a}^b (D(t) - \mu)^2, \mu = \frac{1}{b-a+1} \sum_{t=a}^b D(t)$$

3) 현재 태스크가 Null 태스크이면, 가장 처음 Change Point가 발생하는 시점이 Non-null 태스크로의 전환 시점이 된다.

4) 현재 태스크가 Non-null 태스크이면 특정 Change Point로부터 이후 h 분 동안 어떠한 Change Point도 검출이 되지 않았을 때 이 Change Point는 Null 태스크로의 전환 시점이 된다.

4. 시스템 구현 및 실험

4.1 실험 데이터 구성

본 연구에서 제안한 다중 센서 데이터 기반 태스크 분할 알고리즘의 검증을 위해 분산 미들웨어 기반 세미나실 테스트베드에서 수집된 4개의 사물 센서와 1개의 주변 환경 센서 스트림 데이터를 기반으로 생성된 데이터를 사용하였다. 실제 센서 데이터를 그대로 사용하기에는 센서의 과도한 노이즈와 Non-null 태스크의 길이가 짧고 수가 많지 않다는 한계가 있어, 실제 데이터의 분포를 기반으로 생성한 데이터를 사용하였다.

본 실험에 사용된 CPU는 2.8GHz quad-core Intel Core i7 Processor이다.

4.2 실험 결과 분석

4.2.1 실행 시간 분석

실행 시간 분석의 경우, 센서 데이터 개수를 1000개에서 10,000,000개까지 변화시키며 실행시간을 측정하였다. 실험 결과를 통해 실행 시간은 센서 데이터 개수에 선형 비례함을 알 수 있었는데, 실제로 본 연구에서 제안한 알고리즘의 시간 복잡도는 $O(|D(T_{a,b})|)$ 이다. 10,000,000개의 센서 데이터에 대한 평균 실행 시간은 322.85초로 관찰되었다.

4.2.2 정확도 분석

정확도 분석은 실제 태스크 분할의 관찰 결과와 본 연구에서 제안한 시스템으로부터 얻어진 결과를 비교하여 F1-Score를 계산하였다. 7일 간의 센서 데이터들을 본 시스템에 적용시킨 결과 정상적으로 작동할 경우 평균적으로 93.7%, 비정상적으로 작동할 경우 평균적으로 68.2%의 정확도를 도출하였다.

여러 차례의 실험 결과, 다음과 같은 상황이 발생하여 정확도가 급격히 감소하며 비정상적으로 작동하는 것을 관찰하였다.

1) 시스템 설계에 따르면 h 분 동안 센서 값에 변화가 없었을 경우 Non-null 태스크에서 Null 태스크로의 전환이 일어나는데, 실제로 Non-null 태스크가 진행 중이지만 우연히 h 분 동안 센서 값에 변화가 없어 이를 Null 태스크로의 전환으로 인정한 경우.

2) 시스템 설계에 따르면 시작 지점으로부터 직전 h 분의 데이터를 기준으로 초기 태스크를 판단하는데, 이 결과가 실제 태스크와 다르게 판단이 되는 경우.

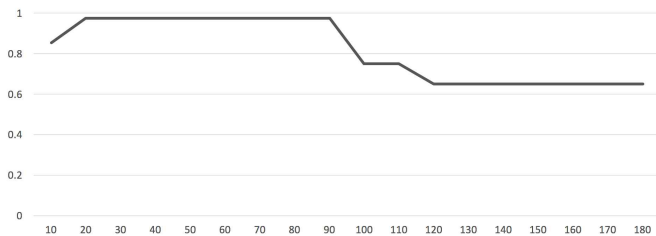


그림 2. h 값에 따른 정확도 변화

그림 2의 그래프에 따르면 h 값에 따라 정확도가 변화하는 것을 관찰할 수 있는데, h 값이 너무 작거나 크면 정확도 하락이 발생했다. 즉, 시스템이 정상적으로 작동하는 h 의 범위가 존재하는 것을 확인하였다.

5. 논의

본 연구의 시스템은 공간의 물리적 구성에 독립적으로 작동하고 다중 센서 데이터를 고려할 수 있으며, 계산량이 적어 실행 시간이 빠르다는 장점이 있다. 하지만, 정확도 분석에서 기술한

내용과 같은 상황이 발생하여 비정상적으로 작동할 가능성이 있는 것으로 나타났다.

첫 번째 문제가 발생하는 이유는 h 값이 짧기 때문인데, h 값이 짧을수록 그 시간 동안 우연히 센서 데이터 변화가 일어나지 않을 가능성이 높기 때문이다. 두 번째 문제가 발생하는 이유는 표본 크기가 너무 작아서 해당 표본이 실제 태스크의 데이터를 대표하지 못하기 때문인데, 통계적으로 보면 표본 크기가 작을수록 노이즈 데이터에 민감하기 때문에 실제 모집단의 평균에서 벗어난 값을 가질 가능성이 높아 정확한 가설 검정이 힘들기 때문이다. 따라서 이러한 문제들을 해결하기 위해서는 h 값을 증가시켜 표본 크기를 늘리고 우연히 센서 데이터의 변화가 발생하지 않을 확률을 낮추는 것이 필요하다.

하지만 본 연구에서 제안한 알고리즘에서는 Non-null 태스크에서 Null 태스크로 전환되는 시점을 h 의 시간동안 센서 데이터 변화가 없는 것으로 정의하기 때문에 h 이하의 시간 간격 동안 발생한 Null 태스크는 인지가 불가하다. 따라서 h 값을 증가시킬수록 인지할 수 있는 Null 태스크의 범위가 줄어들기 때문에 반대로 정확도가 감소하게 된다.

이러한 이유로 h 값의 증가와 감소사이에 정확도의 대립이 발생하기 때문에 적절한 균형점을 찾아 태스크 검출이 정상적으로 작동하게 만드는 과정이 필요하다.

6. 결론

본 연구에서는 기존의 연구방향과 다르게 Change Point Detection을 적용하여 다양한 공간에 확장 가능한 가벼운 (light-weight) 스트림 데이터 태스크 분할 기법을 제안하였다.

실험 결과를 바탕으로 본 연구가 실제 센서 스트림 데이터를 이용한 태스크 분할을 요구하는 스마트 공간에서 즉각적으로 적용될 수 있음을 검증하였다. 하지만 여러 차례의 실험을 거쳐 본 시스템을 특정 스마트 공간에 적용하기 전에 먼저 실험적으로 결정되어야 할 값이 있음을 확인하였다.

본 연구의 확장 방안(Future Work)으로는 해당 스마트 공간에서 미리 수집된 데이터의 패턴 분석을 바탕으로 적절한 h 값의 균형을 찾아내는 알고리즘을 구상할 계획이며, 이러한 알고리즘을 적용하여 더 정확하고 일반적으로 잘 작동하는 시스템을 구축할 계획이다. 또한, 노이즈가 많이 생기는 실제 데이터를 이용한 실험을 통해 시스템 보완에 필요한 부분을 분석하고 개선할 것이다.

7. 시사 문구

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음. [2016-0-00018] 본 연구는 정부(미래창조과학부)의 재원으로 정보통신기술연구진흥센터의 지원을 받아 수행된 연구임. [2017-0-00537, 공간지능을 위한 IoT 사물간 자율협업 기술개발]

8. 참고 문헌

- [1] Okeyo G, Chen L, Wang H, Sterritt R. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. Pervasive and Mobile Computing. 1:10:155-72. 2014 Feb.
- [2] Ye J, Stevenson G. Semantics-driven multi-user concurrent activity recognition. In International Joint Conference on Ambient Intelligence (pp. 204-219). Springer, Cham. 2013 Dec 3.
- [3] Chen L, Nugent CD, Wang H. A knowledge-driven approach to activity recognition in smart homes. IEEE Transactions on Knowledge and Data Engineering. 24(6):961-74. 2012 Jun.
- [4] Laguna JO, Olaya AG, Borrajo D. A dynamic sliding window approach for activity recognition. In International Conference on User Modeling, Adaptation, and Personalization (pp. 219-230). Springer, Berlin, Heidelberg. 2011 Jul 11.
- [5] Hong X, Nugent CD. Partitioning time series sensor data for activity recognition. In Information Technology and Applications in Biomedicine. ITAB 2009. 9th International Conference on (pp. 1-4). IEEE. 2009 Nov 4.
- [6] Truong C, Oudre L, Vayatis N. A review of change point detection methods. arXiv preprint arXiv:1801.00718. 2018 Jan.