

## Ensemble of Nested Dichotomies 기법을 이용한 스마트폰 가속도 센서 데이터 기반의 동작 인지

하으뜸  
부산대학교 전자전기컴퓨터공학과  
(gkdmeema@naver.com)

김정민  
부산대학교 전자전기컴퓨터공학과  
(jeongminkim.islab@gmail.com)

류광렬  
부산대학교 전자전기컴퓨터공학과  
(krroyu@pusan.com)

최근 스마트폰에 다양한 센서를 내장할 수 있게 되었고 스마트폰에 내장된 센서를 이용한 동작 인지에 관한 연구가 활발히 진행되고 있다. 스마트폰을 이용한 동작 인지는 노인 복지 지원이나 운동량 측정, 생활 패턴 분석, 운동 패턴 분석 등 다양한 분야에 활용될 수 있다. 하지만 스마트폰에 내장된 센서를 이용하여 동작 인지를 하는 방법은 사용되는 센서의 수에 따라 단일 센서를 이용한 동작인지와 다중 센서를 이용한 동작인지로 나눌 수 있다. 단일 센서를 이용하는 경우 대부분 가속도 센서를 이용하기 때문에 배터리 부담은 줄지만 다양한 동작을 인지할 때에 특징(feature) 추출의 어려움과 동작 인지 정확도가 낮다는 문제점이 있다. 그리고 다중 센서를 이용하는 경우 대부분 가속도 센서와 중력센서를 사용하고 필요에 따라 다른 센서를 추가하여 동작인지를 수행하며 다양한 동작을 보다 높은 정확도로 인지할 수 있지만 다수의 센서를 사용하기 때문에 배터리 부담이 증가한다는 문제점이 있다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 스마트폰에 내장된 가속도 센서를 이용하여 다양한 동작을 높은 정확도로 인지하는 방법을 제안한다. 서로 다른 10가지의 동작을 높은 정확도로 인지하기 위해 원시 데이터로부터 17가지 특징을 추출하고 각 동작을 분류하기 위해 Ensemble of Nested Dichotomies 분류기를 사용하였다. Ensemble of Nested Dichotomies 분류기는 다중 클래스 문제를 다수의 이진 분류 문제로 변형하여 다중 클래스 문제를 해결하는 방법으로 서로 다른 Nested Dichotomy 분류기의 분류 결과를 통해 다중 클래스 문제를 해결하는 기법이다. Nested Dichotomy 분류기 학습에는 Random Forest 분류기를 사용하였다. 성능 평가를 위해 Decision Tree, k-Nearest Neighbors, Support Vector Machine과 비교 실험을 한 결과 Ensemble of Nested Dichotomies 분류기를 사용하여 동작 인지를 수행하는 것이 가장 높은 정확도를 보였다.

논문접수일 : 2013년 12월 09일    게재확정일 : 2013년 12월 23일

투고유형 : 학술대회우수논문    교신저자 : 류광렬

### 1. 서론

스마트폰의 성능과 기능이 큰 발전을 이룸에 따라 스마트폰 내부의 가속도 센서, 중력 센서, GPS 센서, 자력 센서, 조도 센서, 적외선 센서 등 10여 가지의 다양한 센서들을 내장할 수 있게 되었다. 이에 따라 스마트폰에 내장된 센서를 이용한 많은 연

구들이 진행되고 있는데 그 중 가속도 센서, 중력 센서, GPS 등을 이용하여 인간의 동작을 인지하기 위한 연구 역시 활발히 진행되고 있다. 동작 인지란 특정 센서의 데이터를 이용하여 사용자의 동작을 예측하는 것을 의미하며 이러한 연구는 노인 복지 지원이나 운동량 측정, 생활 패턴 분석, 운동 패턴 분석 등 다양한 분야에 활용하는 것이 가능하다.

\* 본 논문은 산업통상자원부 산업융합원천기술개발사업으로 지원된 연구결과입니다(No.10043907).

하지만 스마트폰에 내장된 센서를 이용하여 동작을 인지하는 경우 스마트폰 기기의 특성상 배터리 용량이 제한되기 때문에 동작 인지에 사용되는 센서의 수를 줄이는 것이 중요 문제가 된다. 하지만 배터리 사용 부담을 줄이기 위해 단일 센서를 사용하면 다양한 동작을 인지할 때의 정확도가 떨어진다는 문제점이 있고 다양한 동작을 높은 정확도로 인지하기 위해 다중 센서를 사용한다면 배터리 사용 부담이 크게 증가하는 문제점이 있다.

따라서 본 논문에서는 배터리 사용 부담을 줄이기 위해 단일 센서를 사용하면서도 다양한 동작을 보다 높은 정확도로 인지할 수 있는 동작 인지 방법을 제안한다. 이를 위해 스마트폰에 내장되어 있는 3축 가속도 센서를 이용하여 데이터를 수집하고 인간의 동작을 앉기, 서기, 걷기, 뛰기를 포함한 기본 동작과 오르막 걷기, 내리막 걷기, 오르막 뛰기, 내리막 뛰기를 포함한 응용 동작, 넘어짐, 다리 절기를 포함한 위험 동작으로 총 10가지 동작을 정의하였다. 그리고 가속도 센서 데이터를 분석하여 각 동작을 인지하는데 유용한 특징을 추출한 후, Ensemble of Nested Dichotomies 분류기를 이용하여 동작 인지를 수행하였다. 분류기의 성능을 검증하기 위해 Decision Tree, k-Nearest Neighbors, Support Vector Machine 세 가지 분류기와 성능을 비교하였다.

## 2. 관련 연구

동작 인지에 관한 연구는 센서의 수에 따라 단일 센서를 이용한 동작인지와 다중 센서를 이용한 동작 인지로 나눌 수 있다.

### 2.1 단일 센서를 이용한 동작 인지

단일 센서를 이용한 연구 중 최근의 연구인 Kose

et al.(2012)에서는 Clustered k-Nearest Neighbors 분류기를 이용하여 실시간으로 동작 인지를 수행하는 방법을 제안하였다. k-Nearest Neighbors를 수행하기 전 사전에 클러스터를 형성하여 계산량을 줄여 배터리 효율은 높아 졌지만 높은 정확도로 인지 가능한 동작의 수가 네 가지로 제한되고 정확도는 92.13%였다.

다른 최근의 연구인 Ataya and Jallon(2012)에서는 Support Vector Machine과 Markov Chain을 이용한 동작 인지에 이용하였다. Support Vector Machine을 이용하여 동작을 인지한 후, Markov Chain을 이용하여 정확도를 향상시키는 방법을 제안하였다. 분류 가능한 동작은 눕기, 앉기, 구부정하게 앉기, 서기, 걷기, 뛰기이고 정확도는 92%였다.

단일 센서를 사용한 기존 연구의 경우 가속도 센서만 사용하기 때문에 배터리 사용 부담은 줄지만 높은 정확도로 인지 가능한 동작의 수가 제한되고 인지 정확도가 다중 센서에 비해 낮다는 문제점이 있다.

### 2.2 다중 센서를 이용한 동작인지

다중 센서를 이용한 최근 연구인 Cho et al.(2012)에서는 가속도 센서, 중력 센서, 자력 센서를 사용하였고 SVM 분류기를 사용하여 동작인지를 수행하였다. 다수의 센서를 사용하기 때문에 배터리 사용 부담은 늘지만 높은 정확도로 인지 가능한 동작은 부동 자세, 걷기, 계단 올라가기, 계단 내려가기, 뛰기로 늘어났고 인지 정확도는 98.26%였다.

다른 최근의 연구인 Angguita et al.(2012)에서는 가속도 센서, 중력 센서를 이용하였고 총 17개의 특징을 생성하였으며 Hardware Friendly-SVM 분류기를 사용하였다. 두 개의 센서 사용과 다수의 특징을 사용함으로 인한 배터리 부담을 줄이기 위

해 Hardware Friendly-SVM을 사용하였으나 앉기, 서기, 눕기, 뛰기, 계단 올라가기, 계단 내려가기 동작에 대한 정확도가 89%였다.

다중 센서를 사용하는 경우 단일 센서에 비해 다양한 동작을 높은 정확도로 인지할 수 있지만 배터리 부담이 크다는 문제점이 있었다.

### 3. 동작 인지

#### 3.1 가속도 센서 데이터 수집

3축 가속도 센서란 단위 시간당 속도의 변화를 검출하는 소자로, 가속도, 진동, 충격 등의 동적인 힘을 감지하며 관성력, 전기 변형, 자이로의 원리를 이용한 것이다.



<Figure 1> Samsung Galaxy Note II

가속도 센서 데이터를 수집하기 위해 Android 4.1버전 기반의 Samsung Galaxy Note II를 사용하였다. 데이터는 5명의 지원자(24~26세)에게 각 동작을 2분씩 수행하도록 하여 0.1초 간격으로 센서 데이터를 수집하였고 기기는 하의 앞주머니에 액정이 정면을 향하도록 넣고 데이터 수집을 수행하였다. 각 동작 별로 10분간의 데이터를 수집하였고 특징을 생성 할 때는 기준 시점을 포함한 과거

2초간의 데이터를 이용한다. 하지만 스트림 데이터의 특성상 첫 2초 동안은 시간 창(time window) 내에 과거 데이터가 충분하지 않기 때문에 특징을 생성하는 것이 불가능하다. 따라서 첫 2초 동안의 데이터를 제외하고 각 동작 별로 5900개(=  $5 \times (60 \times 2 - 2) / 0.1$ )의 데이터를 수집하였다. 그리고 각 동작 별로 4700개의 데이터를 추출하여 학습에 사용하였고 나머지 1200개의 데이터를 성능 평가를 위해 사용하였다.

#### 3.2 동작 정의

인간의 동작은 무수히 많이 때문에 동작 인지에 필요한 유용한 동작을 정의할 필요가 있다. 따라서 본 논문에서는 사람이 운동할 때 취하는 동작 중 가장 빈도가 높은 10가지를 기본 동작 4가지와 기본 동작을 응용한 4가지 동작, 위험 동작 2가지로 정의 하였다. <Table 1>은 10가지 동작에 대한 정의를 보인 것이다.

<Table 1> Activities to be Distinguished

Activity Number	Description
1	Sitting
2	Standing
3	Walking on flat road
4	Running on flat road
5	Walking on uphill road
6	Walking on downhill road
7	Running on uphill road
8	Running on downhill road
9	Falling
10	Hobbling

#### 3.3 학습을 위한 특징 생성

가속도 센서로 수집된 원시 데이터만으로 분류

기를 학습하여 동작을 인지하는 것은 한계가 있다. 그러므로 각 동작의 데이터를 분석하여 분류기 학습에 유용한 특징을 추출할 필요가 있다. 따라서 본 논문에서는 시간 창 내에서 17가지의 특징을 생성하였다.

$$TiltAngle(x) = atan\left(\frac{x}{\sqrt{y^2+z^2}}\right) \quad (1)$$

$$Mag = \sqrt{x^2+y^2+z^2} \quad (2)$$

식 (1)은 x축에 대한 Tilt angle을 구하는 식이고 식 (2)는 벡터의 크기를 구하는 식이다. 두 식에서  $x, y, z$ 는 가속도 센서 데이터의 각 축을 의미한다. <Table 2>는 학습 및 분류에 사용되는 특징들에 대한 설명을 정리한 것이다.

<Table 2> Descriptions of Features

Maximum acceleration in each direction in the time window of the last 2 seconds
Minimum acceleration in each direction in the time window of the last 2 seconds
Maximum tilt angle of each axis in the time window of the last 2 seconds
Minimum tilt angle of each axis in the time window of the last 2 seconds
Magnitudes of acceleration vector at each time point
Maximum magnitude of acceleration vector in the time window of the last 2 seconds
Minimum magnitude of acceleration vector in the time window of the last 2 seconds
Difference between the maximum and the minimum magnitude of acceleration vector in the time window of the last 2 seconds
Standard deviation of the magnitudes of acceleration vector in the time window of the last 2 seconds

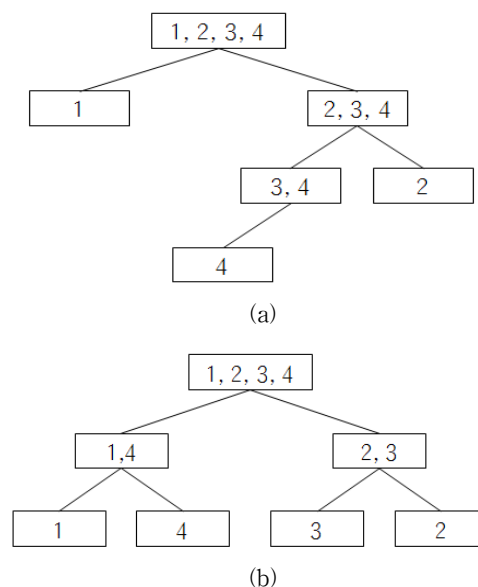
### 3.4 동작 인지를 위한 분류기

분류기의 성능이 동작 인지의 정확도에 큰 영향

을 미치기 때문에 동작 인지를 위해서는 다중 클래스 문제에 적합한 분류기를 선택하는 것이 중요한 문제가 된다. 따라서 본 논문에서는 Ensemble of Nested Dichotomies 분류기를 사용하였으며 적합성을 검증하기 위해 Decision Tree, k-Nearest Neighbors, Support Vector Machine과 성능을 비교하였다.

#### • Ensemble of Nested Dichotomies(END) :

END 분류기는 Nested Dichotomy(ND) 기반의 앙상블 학습 방법이다. ND는 내부 분류기를 이용하여 이진트리를 생성하고 END는 ND를 이용하여  $n$  개의 각각 다른 이진트리를 무작위로 생성하여 각 트리의 분류 결과를 토대로 투표를 통해 멀티 클래스 문제를 해결하는 기법이다. 루트 노드에는 모든 클래스를 가지고 있고 트리를 확장시킬 때 이진 분류기를 이용하여 임의의 두 부분집합(subset)



<Figure 2> Two Different Versions of Nested Dichotomies

으로 나누어 나간다. 트리의 확장은 마지막 노드(leaf node)에 하나의 클래스만 남을 때까지 계속한다. 이렇게 서로 다른 모양의 트리를  $n$ 개 생성한다. 여기서는  $n$ 은 임의로 설정하여 학습을 수행한다(Frank and Kramer, 2004). 본 논문에서는 실험을 통해 내부 분류기를 Random tree를 사용하였고  $n$ 은 10으로 설정하였다. <Figure 2>는 ND의 다른 두 형태를 나타낸 것이다.

- **Decision Tree(DT)** : DT는 어떤 의사 결정을 내리기 위한 규칙을 트리 형태로 구성한 것으로 데이터마이닝 분석의 대표적인 분석 방법 중 하나이다. DT를 학습하는 여러 가지 알고리즘이 있고 그중 대표적인 알고리즘으로 C4.5가 있다. DT 학습에 C4.5를 사용하는 경우, 노드를 결정하기 위해 후보 속성(Attribute)들의 정보이득을 계산하여 정보이득이 가장 큰 속성을 노드로 결정한다. 이러한 과정을 말단 노드에 하나의 클래스가 남을 때까지 반복한다. 정보이득이란 노드를 추가하기 전의 데이터 복잡도와 노드를 추가한 후의 데이터 복잡도의 차를 통해 구할 수 있고 그 차가 클수록 분류가 잘 된 것이다(Russell and Norvig, 2010).

- **$k$ -Nearest Neighbors( $k$ -NN)** :  $k$ -NN은 기존에 가지고 있는 데이터를 이용하여 별도의 학습 과정 없이 새로운 데이터를 분류하는 기법이다. 입력 데이터로부터 기존 데이터 내에서 가까운  $k$ 개의 데이터가 갖는 클래스를 이용하여 투표를 통해 새로운 데이터의 클래스를 분류한다. 데이터 사이의 거리는 계산하는 방법으로는 유클리디안 거리 혹은 맨해튼 거리 계산 방법을 이용한다(Russell and Norvig, 2010).

- **Support Vector Machine(SVM)** : SVM은 두 클래스를 분류하기 위한 분류 방법으로 데이터 마이닝 분야에서 가장 많이 사용되는 분류기이다. SVM은 벡터 공간에서 두 클래스를 나누기 위한 Hyper plane으로부터 가장 가까운 Support vector를 찾는다. 그리고 Support vector 사이의 거리(Margin)가 최대가 되는 Hyper plane을 찾음으로서 두 클래스를 분류하는 것이다. SVM은 두 클래스를 나누기 위한 분류기이기 때문에 다중 클래스 분류하기 위해서는 1-vs-rest 혹은 1-vs-1 방법을 사용한다. 본 논문에서는 1-vs-rest 방법을 사용하였다(Russell and Norvig, 2010).

<Table 3> Accuracy of END for Activity Recognition(the Numbers Used for Column and Row Headings Indicate the Activities Shown in <Table 1>)

	1	2	3	4	5	6	7	8	9	10	Accuracy(%)
1	2000	0	0	0	0	0	0	0	0	0	100.0
2	0	1200	0	0	0	0	0	0	0	0	100.0
3	0	4	1996	0	0	0	0	0	0	0	99.7
4	0	0	0	1194	5	1	0	0	0	0	99.5
5	0	0	2	0	1067	31	0	0	53	47	88.9
6	0	0	16	0	0	1181	0	0	0	3	98.4
7	0	0	0	0	1	0	1198	0	0	1	99.8
8	0	0	0	0	0	0	0	1200	0	0	100.0
9	0	0	0	11	0	0	0	0	1189	0	99.1
10	0	0	0	0	0	0	0	0	15	1185	94.4
Average											98.4

## 4. 실험 및 결과 분석

본 논문에서 제안한 동작 인지 방법의 성능을 평가하기 위해 Android 4.1버전 기반의 Samsung Galaxy Note II를 이용하여 실험을 하였다. 실험은 END 분류기의 동작 인지 정확도를 확인하기 위한 실험과 END 분류기의 타당성을 검증하기 위해 다른 분류기와의 성능 비교를 수행하였다.

첫 번째 실험은 END의 동작인지 정확도를 확인한 실험으로 결과는 <Table 3>과 같다. 앉기와 서기와 같은 정적인 동작을 제외한 나머지 동작 중 내리막길 뛰기가 동작이 100%로 정확도가 가장 높았고 오르막 길 걷기 동작이 88.9%로 정확도가 가장 낮았다. 오르막 길 걷기 동작의 경우 내리막 길 걷기와 넘어짐, 달리 절기 동작으로 잘 못 분류 하는 경우가 있었다.

두 번째 실험은 다른 분류기와의 정확도 비교를 수행하였다. <Table 4>는 각 분류기의 동작인지 정확도를 비교한 것이다. DT 학습에는 C4.5 알고리즘을 사용하였고  $k$ -NN의  $k$ 는 실험을 통해 10으로 설정하였으며 SVM의 커널은 Polynomial kernel을 사용하였다. 실험 결과 END를 이용하여 동작인지를 하는 것이 가장 높은 정확도를 보였고  $k$ -NN이 가장 낮은 정확도를 보였다.

<Table 4> Comparison of Accuracies of Different Classifiers

Classifier	Accuracy(%)	Note
DT	97.6	C4.5 algorithm
$k$ -NN	96.5	$k = 10$
1-vs-rest SVM	97.6	Max margin = 40 with polynomial kernel
END	<b>98.4</b>	Random forest

## 5. 결론

본 논문에서는 스마트폰 상에서 가속도 센서를 사용하여 다양한 동작을 높은 정확도로 인지를 하기 위한 방법을 제안하였다. 자원의 제약이 있는 스마트폰에서 배터리 부담을 줄이기 위해 가속도 센서만을 이용하여 데이터를 수집하고 10가지 동작의 센터 데이터를 분석하여 유용한 17가지 특징을 추출하였다. 그리고 다양한 동작의 인지 정확도를 향상시키기 위해 END 분류기를 사용하여 각 동작을 분류하였다. 실험을 통해 확인한 결과 END 분류기를 사용하여 동작 인지를 수행하는 것이 DT 분류기를 사용하는 것 보다 평균 0.8%의 정확도 향상 되었고  $k$ -NN보다는 평균 1.9%, SVM 분류기 보다는 평균 0.8%의 정확도가 향상 되었다.

## 참고문헌

- Anguita, D., A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," *Proceeding IWAAL'12 Proceedings of the 4th international conference on Ambient Assisted Living and Home Care*, (2012), 216~223.
- Ataya, A. and P. Jallon, "Amelioration of physical activity estimation from accelerometer sensors using prior knowledge," *Signal Processing Conference(EUSIPCO), 2012 Proceedings of the 20th European*, (2012), 954~958.
- Cho, J. H., J. T. Kim and T. S. Kim, "Smart Phone-based Human Activity Classification and Energy Expenditure Generation in Building Environments," *SHB2012-7th International Symposium on Sustainable Healthy*

- Buildings*, (2012), 97~105.
- Frank, E. and S. Kramer, "Ensembles of nested dichotomies for multiclass problems," *Proceeding ICML '04 Proceedings of the twenty-first international conference on Machine learning*, (2004), 39~46.
- Kose, M., O. Incel and C. Ersoy, "Online Human Activity Recognition on Smart Phones," *In Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*, (2012), 11~15.
- Russell, S. J. and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3<sup>rd</sup> edition, Prentice Hall, Englewood Cliffs, New Jersey, 2010.

Abstract

## Ensemble of Nested Dichotomies for Activity Recognition Using Accelerometer Data on Smartphone

Eu Tteum Ha\* · Jeongmin Kim\* · Kwang Ryel Ryu\*\*

As the smartphones are equipped with various sensors such as the accelerometer, GPS, gravity sensor, gyros, ambient light sensor, proximity sensor, and so on, there have been many research works on making use of these sensors to create valuable applications. Human activity recognition is one such application that is motivated by various welfare applications such as the support for the elderly, measurement of calorie consumption, analysis of lifestyles, analysis of exercise patterns, and so on. One of the challenges faced when using the smartphone sensors for activity recognition is that the number of sensors used should be minimized to save the battery power. When the number of sensors used are restricted, it is difficult to realize a highly accurate activity recognizer or a classifier because it is hard to distinguish between subtly different activities relying on only limited information. The difficulty gets especially severe when the number of different activity classes to be distinguished is very large.

In this paper, we show that a fairly accurate classifier can be built that can distinguish ten different activities by using only a single sensor data, i.e., the smartphone accelerometer data. The approach that we take to dealing with this ten-class problem is to use the ensemble of nested dichotomy (END) method that transforms a multi-class problem into multiple two-class problems. END builds a committee of binary classifiers in a nested fashion using a binary tree. At the root of the binary tree, the set of all the classes are split into two subsets of classes by using a binary classifier. At a child node of the tree, a subset of classes is again split into two smaller subsets by using another binary classifier. Continuing in this way, we can obtain a binary tree where each leaf node contains a single class. This binary tree can be viewed as a nested dichotomy that can make multi-class predictions. Depending on how a set of classes are split into two subsets at each node, the final tree that we obtain can be different. Since there can be some classes that are correlated, a particular tree may perform better

---

\* Department of Electrical and Computer Engineering, Pusan National University

\*\* Corresponding Author: Kwang Ryel Ryu

Department of Electrical and Computer Engineering, Pusan National University  
Busandaehak-ro-63beon-gil, Geumjeong-gu, Busan 609-735,  
Tel: +82-51-510-2453, Fax: +82-51-517-2431, E-mail: hongkd@university.ac.kr



than the others. However, we can hardly identify the best tree without deep domain knowledge. The END method copes with this problem by building multiple dichotomy trees randomly during learning, and then combining the predictions made by each tree during classification. The END method is generally known to perform well even when the base learner is unable to model complex decision boundaries. As the base classifier at each node of the dichotomy, we have used another ensemble classifier called the random forest. A random forest is built by repeatedly generating a decision tree each time with a different random subset of features using a bootstrap sample. By combining bagging with random feature subset selection, a random forest enjoys the advantage of having more diverse ensemble members than a simple bagging. As an overall result, our ensemble of nested dichotomy can actually be seen as a committee of committees of decision trees that can deal with a multi-class problem with high accuracy.

The ten classes of activities that we distinguish in this paper are ‘Sitting,’ ‘Standing,’ ‘Walking,’ ‘Running,’ ‘Walking Uphill,’ ‘Walking Downhill,’ ‘Running Uphill,’ ‘Running Downhill,’ ‘Falling,’ and ‘Hobbling.’ The features used for classifying these activities include not only the magnitude of acceleration vector at each time point but also the maximum, the minimum, and the standard deviation of vector magnitude within a time window of the last 2 seconds, etc. For experiments to compare the performance of END with those of other methods, the accelerometer data has been collected at every 0.1 second for 2 minutes for each activity from 5 volunteers. Among these 5,900 ( $= 5 \times (60 \times 2 - 2) / 0.1$ ) data collected for each activity (the data for the first 2 seconds are trashed because they do not have time window data), 4,700 have been used for training and the rest for testing. Although ‘Walking Uphill’ is often confused with some other similar activities, END has been found to classify all of the ten activities with a fairly high accuracy of 98.4%. On the other hand, the accuracies achieved by a decision tree, a k-nearest neighbor, and a one-versus-rest support vector machine have been observed as 97.6%, 96.5%, and 97.6%, respectively.

**Key Words** : Ensemble of Nested Dichotomy, Activity Recognition, Smartphone Accelerometer

## 저 자 소개



**하으뜸**

신라대학교 컴퓨터정보공학부 학사 졸업하여, 현재는 부산대학교 전자전기컴퓨터공학과 석사 과정중이며, 관심 연구분야는 영상처리, 기계 학습, 진화형 시스템, 데이터 마이닝, 확률적 추론 등이다.



**김정민**

부산대학교 컴퓨터공학과 학사 졸업 후 현재는 부산대학교 전자전기컴퓨터공학과 박사 과정중이며, 관심 연구 분야는 기계 학습, 진화형 시스템, 데이터 마이닝, 확률적 추론 등이다.



**류광렬**

현재 부산대학교 전자전기컴퓨터공학과 교수로 재직 중이다. 서울대학교 전자공학과 학사 졸업하여 동대학 전자공학과 석사학위 후 미시간대학교 전자전기컴퓨터공학과 박사과정 취득하였다. 관심 연구분야는 기계 학습, 데이터 마이닝, 진화형 시스템, 확률적 추론 등이다.