Particle Swarm Optimization Using Adaptive Boundary Correction for Human Activity Recognition

Yongjin Kwon[1], Seonguk Heo[2], Kyuchang Kang[1], and Changseok Bae[1]
[1] SW•Content Research Laboratory, Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Republic of Korea
[e-mail: {scocso, k2kang, csbae}@etri.re.kr]
[2] Wellness Convergence Research Center, Daegu Gyeongbuk Institute of Science and Technology
333, Techno jungang-daero, Hyeonpung-myeon, Dalseong-gun, Daegu, 711-833, Republic of Korea
[e-mail: h75304@dgist.ac.kr]
*Corresponding author: Changseok Bae

Received April 5, 2014; revised May 28, 2014; accepted June 3, 2014; published June 27, 2014

Abstract

As a kind of personal lifelog data, activity data have been considered as one of the most compelling information to understand the user's habits and to calibrate diagnoses. In this paper, we proposed a robust algorithm to sampling rates for human activity recognition, which identifies a user's activity using accelerations from a triaxial accelerometer in a smartphone. Although a high sampling rate is required for high accuracy, it is not desirable for actual smartphone usage, battery consumption, or storage occupancy. Activity recognitions with well-known algorithms, including MLP, C4.5, or SVM, suffer from a loss of accuracy when a sampling rate of accelerometers decreases. Thus, we start from particle swarm optimization (PSO), which has relatively better tolerance to declines in sampling rates, and we propose PSO with an adaptive boundary correction (ABC) approach. PSO with ABC is tolerant of various sampling rate in that it identifies all data by adjusting the classification boundaries of each activity. The experimental results show that PSO with ABC has better tolerance to changes of sampling rates of an accelerometer than PSO without ABC and other methods. In particular, PSO with ABC is 6%, 25%, and 35% better than PSO without ABC for sitting, standing, and walking, respectively, at a sampling period of 32 seconds. PSO with ABC is the only algorithm that guarantees at least 80% accuracy for every activity at a sampling period of smaller than or equal to 8 seconds.

Keywords: Activity recognition, lifelog, sampling rate, particle swarm optimization (PSO), adaptive boundary correction (ABC)

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2010-0028631).
A preliminary version of this paper was presented at ICONI 2013 and was selected as an outstanding paper.

http://dx.doi.org/10.3837/tiis.2014.06.015

1. Introduction

Personal life information is currently being treated as an important resource for more accurate diagnoses. Practitioners need routine data, as well as vital data, to see patient's life patterns for precise, predictive, and personalized treatments. For this reason, there have been many studies on using routine data to help practitioners. Ito et al. [1] tried to overcome the limitation of medical information by gathering voluminous and varied lifelog data for medical purposes. Heo et al. [2] developed an Android-based smartphone application to encourage breast self-examination in daily life to find breast cancer at an early stage. Kwon et al. [3] built a lifelog agent based on Health Avatar Platform [4] to summarize human activity patterns using smartphone sensors in order to incorporate lifelog data into a medical record.

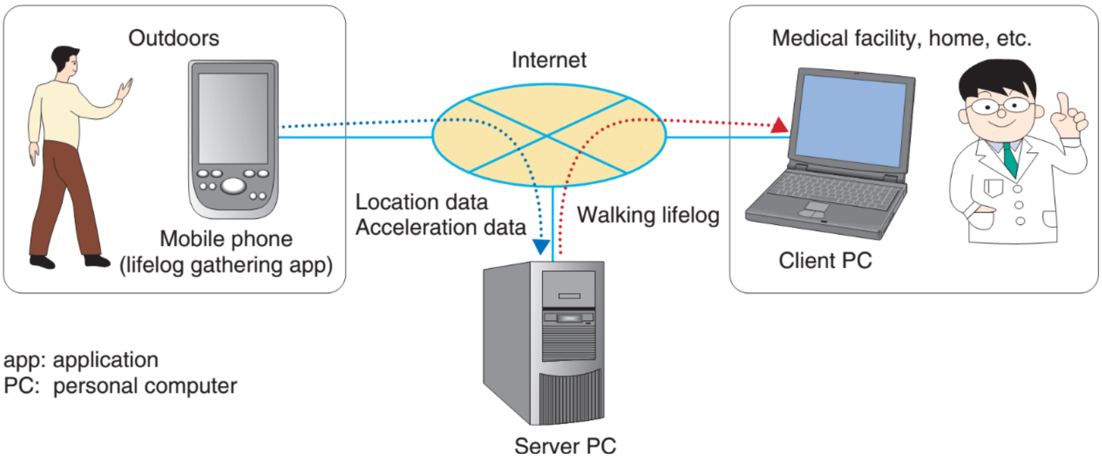

Fig. 1. System dialog for medical diagnosis and treatment with lifelog in NTT [1]

A lifelog is all information that can be collected from a person, such as sleep information, exercise information, and eating habits. A lifelog can be collected from a variety of devices such as smartphones or wearable sensors, and exploited for various purposes, including human activity recognition. For instance, a digital health screening form [5-6] requires lifestyle information such as activity, sleeping, exercising, and eating habits, as well as entries in a paper-based health screening form. If questions such as "how long do you walk per day," "how long do you sit at a desk each day," or "how often do you exercise" can be filled in with activities automatically derived from the lifelog data, it is then possible to obtain the objective lifestyle information. Hence, practitioners can easily achieve a more accurate diagnosis using a digital health screening form.

In particular, activity data are one of the most fascinating information retrieved from the lifelogs. Activity is actually something that is done at some time (e.g. sitting, walking). Activity data in daily routines have been used to understand the user's habits or lifestyle, and have been used in some applications. For instance, Zwartjes et al. [7] developed an ambulatory monitoring system that analyzes current activity of the patient to monitor motor symptoms in Parkinson's disease. Another example is using activity data collected from smartphone sensors to measure the amount of calories burned [3].

In this paper, we choose lifelog data as the sensor values of a triaxial accelerometer in a smartphone, and propose a robust method to sampling rates that recognizes the user's activities. Since lifelog data are collected over a long period of time, it is necessary to pay attention to actual smartphone usage, battery consumption, or storage use of a smartphone, which highly depends on a sampling rate of the accelerometer, because a high sampling rate leads to high accuracy in activity recognition. Activity recognition methods with well-known classification algorithms, including MLP [8], C4.5 [9], or SVM [10], suffer from a loss of accuracy when the sampling rate decreases. Thus it is not suitable to implement activity recognition by direct adaptions of typical classification algorithms.

We therefore introduce particle swarm optimization (PSO) [11-12] using adaptive boundary correction (ABC). PSO is a stochastic method that tries to repeatedly enhance a particle (candidate solution) of a given problem. One of the greatest advantages is that in PSO, each particle's movement is affected by the best known position of the entire group as well as by the best known position of the particle, which makes the particle move to the more optimal position. We start from PSO for classification, described in [13], that has better tolerance to decreases in sampling rates. Using PSO for classification, however, may fail to classify some input data for a low sampling rate. The ABC method helps recognize the unclassified points by adjusting the classification boundary based on the shape of the input data.

The rest of this paper is organized as follows. In section 2, the related work on human activity recognition and particle swarm optimization is summarized. Section 3 describes the ABC technique for activity classification. Section 4 provides an analysis of accelerations to see the properties of each activity. The experimental results are then shown in section 5, and we offer some concluding remarks in section 6.

2. Related Work

2.1 Human activity recognition

Human activity recognition detects the person's activities with an analysis of the acceleration data. It has been studied in computer vision, wearable computing, data mining, and their applications. Although there are some studies [14-16] to detect micro activities with vision systems, we concentrated on accelerometers that are small, light-weight, and embedded in a large number of portable devices. Since accelerometers measure the change in a person's body, they play a significant role in activity recognition. To determine the human activities properly, at least one accelerometer must be attached to the body so that acceleration data of the person are recorded during certain activities. The acceleration data are converted into statistic data after feature extraction, and are then applied to algorithms for activity recognition.

Some applications apply human activity recognition for several purposes. Jafari et al. [17] used accelerometers to perceive the urgent states of elderly users. These allow others to be alerted when such users face a potentially dangerous situation. The authors in [18-20] exploited activity recognition for activity monitoring, and transform exercise information into user-friendly information after classification. Kwon et al. [3] incorporated the activity recognition with healthcare service platform to provide an application to show the user's activity patterns in a graph.

Several studies have introduced algorithms and models for activity recognition. Stikic et al. [21] used RFIDs attached to objects used at home, along with accelerometers, to record what the person is doing and what object is used each time. Some studies [22-23] have developed algorithms for human activity recognition, each of which is concentrated on a distinct collecting method or different target activities. Uddin et al. [24] used body joint-angle features and a hidden Markov model for activity recognition with a single camera.

2.2 Particle swarm optimization

Particle swarm optimization (PSO) is one of the best algorithms among swarm-based approaches that are reasonable ways to solve a number of difficult problems in nature. PSO, inspired by flocks of birds and schools of fishes, is a stochastic method that iteratively improves a candidate solution of a given problem to find the optimal solution [11-12]. In PSO, every particle shares its social experience derived from traveling. Depending on the collective experience, the entire group moves to the optimal position.

In PSO, each particle computes its next velocity using the states of the other particles, and then computes its next position. After the particle changes its position, it shares its position with the others, helping them determine their next positions. The process is repeated until an optimal position is found.

$$v^{t+1} = w \times v^t + c_1 \times rand() \times (pBest - x^t) + c_2 \times rand() \times (gBest - x^t) \quad (1)$$

$$x^{t+1} = x^t + v^{t+1} \quad (2)$$

The velocity and next position of a particle can be computed through equations (1) and (2), where v^t and x^t indicate the velocity and position of the particle at time t, respectively. In this paper, constants w, c_1, and c_2 are chosen to be 1, 2, and 2, respectively. In addition, $rand()$ generates a real number within [0, 2], $pBest$ is the best-known position of a particle, and $gBest$ is the best-known position of the entire group. PSO moves the particles by iterating equations (1) and (2) to find the optimal position of the group. PSO is finished when the termination criterion is met, for instance, PSO is repeated for a given number of times, or until the next velocity is zero.

3. Adaptive Boundary Correction

3.1 Data classification using a boundary

In classification, the input data can be assigned to a class using a set of representative positions of each class, which can be determined by PSO. To perform classification quickly and accurately, we first create the classification boundaries of each class, as described in [13]. For each feature, the boundary of a class is computed using the equations (3) and (4).

$$LowerBound = x_i - rand() \times (Max_i - Min_i) \quad (3)$$

$$UpperBound = x_i + rand() \times (Max_i - Min_i) \quad (4)$$

where x_i is the i-th feature value of a representative position, $rand()$ generates any number within [0,1], Max_i is the maximum value among the i-th feature values, and Min_i is the minimum value among the i-th feature values. After creating the boundary, one can determine whether a given point is a member of the class using the boundary, as shown in **Fig. 2**.

Fig. 3 describes a pseudo code of data classification using a boundary. For each feature value of a given point, if it is not included in the boundary, we then neglect the point because

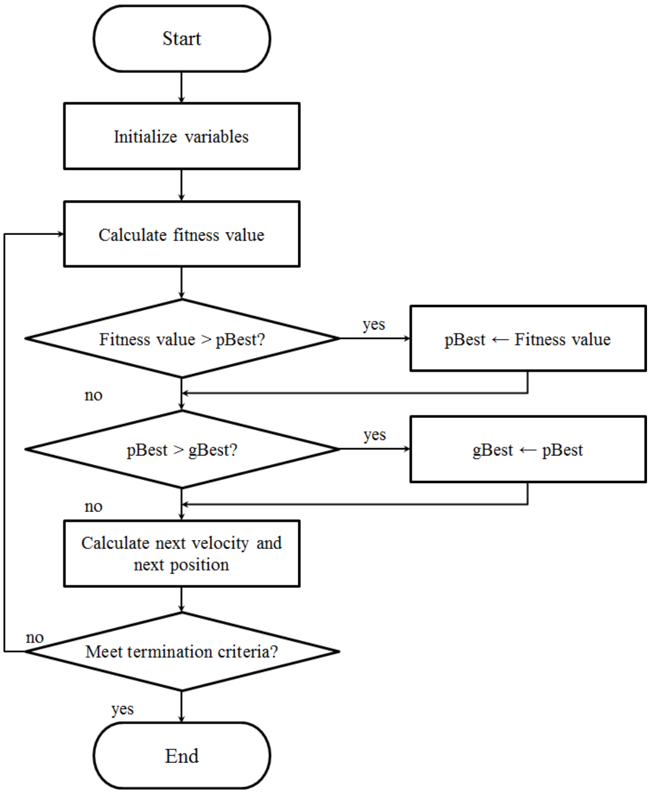

Fig. 2. Flowchart of particle swarm optimization

it is not a member of the class. If all of the feature values are included in the boundaries, we can conclude that the point is a member of the class.

```
for i ← 0 to n
    if Lower_i ≤ x_i and Upper_i ≥ x_i then
        continue
    else
        return false
    end if
end for
return true
```

Fig. 3. The pseudo code of data classification using a boundary

3.2 Challenges of classification using a boundary

Classification using a boundary has the advantage of being executed quickly and accurately. However, the method has some drawbacks. Since the feature values are likely to vary when a sampling rate becomes lower, the test points may not be fit in the boundary of the class generated by the training data, even though they are actually members of the class. For instance, consider that a class is distributed within $[1, 10]$ and its representative point is 5. From equations (3) and (4), the boundary of the class can be computed as $LowerBound = -4$ and $UpperBound = 14$ (if $rand() = 1$). If the test data are positioned within $[-5, 15]$, a point out of $[1, 10]$ should be determined as not being a member of the class. Since the point is the actual member, this situation causes a classification error. Another example is shown in **Fig. 4**. Although point F should be a member of class B, the algorithm does not assign F to class B because it is located outside the boundary of class B.

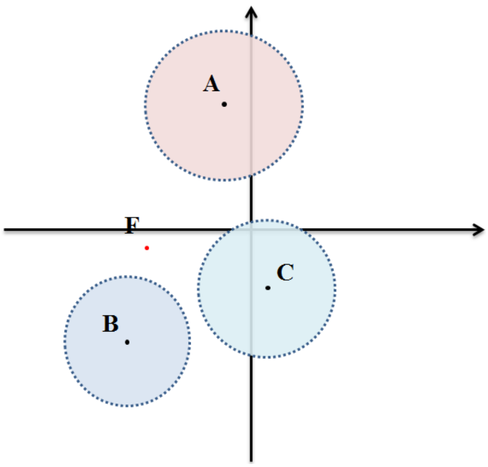

Fig. 4. Example of a point outside the boundary of a class

Points outside the boundary are sometimes found during activity classification. In general, the training data are collected intentionally under noiseless conditions. The test data, however, are recorded under diverse and noisy conditions. In particular, the sampling rate of the accelerometer plays a significant role in activity recognition [25]. Since these points are not able to be classified, they cause a decrease in the classification accuracy. A method to cope with such points to reduce the classification errors should be derived.

3.3 Adaptive boundary correction (ABC) approach

A naïve approach to solve the out-of-boundary problem is used to expand the boundary of each class. As shown in **Fig. 5**, however, the extension of the boundaries may cause an intersection of the regions of two classes. The intersection may cause another problem, i.e., which cluster the point in the overwrapped region is assigned to.

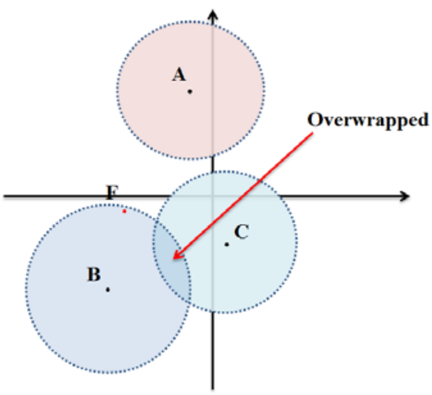

Fig. 5. Problem of a naive approach

We propose an Adaptive Boundary Correction (ABC) approach to solve the out-of-boundary problem without overwrapped regions. The ABC approach copes with singular points outside the boundaries of every class by applying another method to increase the classification accuracy. In this paper, we used the distance from each representative point to the singular point. Since the representative points are known after the PSO phase, we can compute the distance from each representative point, and can find which representative point is the closest one. The singular point can then be determined as a member of the corresponding class that has the closest representative point. **Fig. 6** shows the flow of the ABC approach.

For instance, assume that there is a point F that is assigned to no group, as F is not within the boundary of any class, as shown in **Fig. 7**. According to the ABC approach, to assign F to one of the three classes, we first need to compute the distances $\overline{AF}$, $\overline{BF}$, and $\overline{CF}$ as follows:

$$\overline{AF} = \sqrt{(a_1 - f_1)^2 + (a_2 - f_2)^2 + \cdots + (a_i - f_i)^2} \tag{5}$$

$$\overline{BF} = \sqrt{(b_1 - f_1)^2 + (b_2 - f_2)^2 + \cdots + (b_i - f_i)^2} \tag{6}$$

$$\overline{CF} = \sqrt{(c_1 - f_1)^2 + (c_2 - f_2)^2 + \cdots + (c_i - f_i)^2} \tag{7}$$

where i is the number of features, $F = (f_1, f_2, \cdots, f_i)$, and the representative points of each class are given as $A = (a_1, a_2, \cdots, a_i)$, $B = (b_1, b_2, \cdots, b_i)$, and $C = (c_1, c_2, \cdots, c_i)$. Comparing these distances, we can find the closest class, B. Hence, we determine that F is a member of the class B.

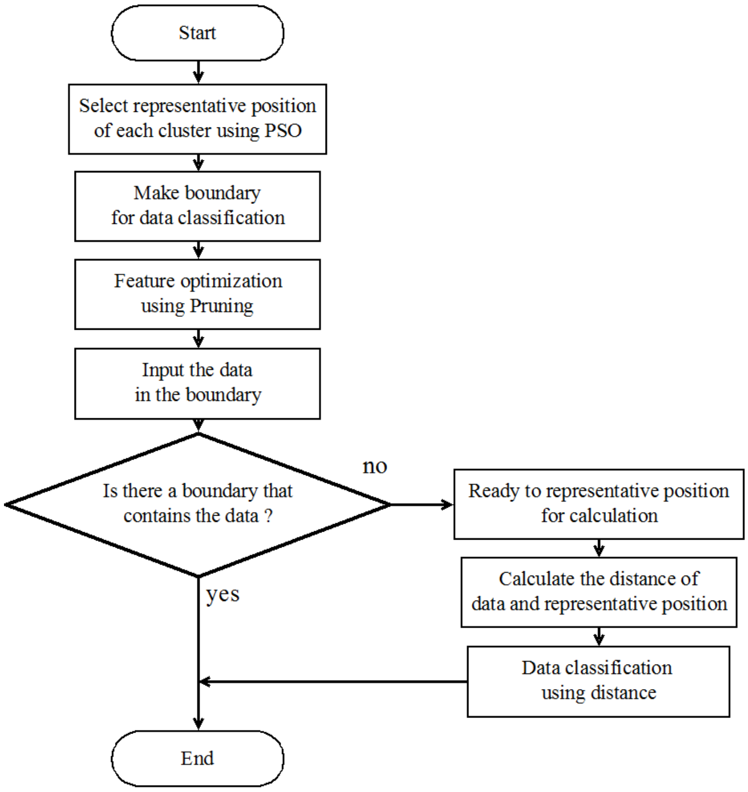

Fig. 6. Flowchart of the ABC approach

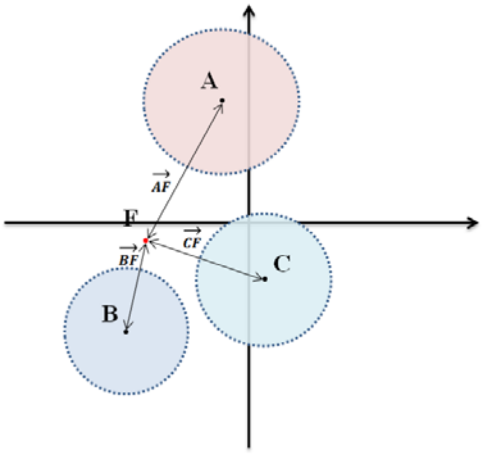

Fig. 7. Example of ABC approach

4. Analysis of Activity Data

4.1 Data collection

To evaluate the PSO using ABC for activity recognition, we collect the activity data in real life. Since a smartphone has a built-in triaxial accelerometer, this experiment exploits a smartphone by implementing an application that handles the sensor. A smartphone was kept in a pants pocket to collect triaxial acceleration during a series of activities. The sensor data were recorded in a database and used to verify the effectiveness of the proposed algorithm.

Fig. 8 shows the position of the smartphone and axes of the accelerometer relative to the user. The x axis implies the horizontal movement of the user, the y axis implies the vertical movement, and the z axis implies the movement of the leg. The unit of acceleration is m/s^2.

Fig. 8. Direction of the accelerometer

4.2 Properties of the activities

Before analyzing the sensor data of a user, we need to determine activities to be recognized. In general, the basic human activities during a daily routine are walking, sitting, and standing. To describe the formal description of each activity, we referred to the definitions in [26]. The formal definition of the three activities are presented as follows:

- Sitting : Sitting down inactively. Does not include working on computer or reading while sitting
- Standing : Standing without moving of legs
- Walking : Walking without carrying any items in your hand or on your back heavier than a pound

These activities are mutually exclusive, and are considered to be the most basic activities in many studies. To identify these activities, it is necessary to know the properties of each activity that are exploited during the activity recognition processes. In this subsection, we analyze the real sensor data and compute the statistics of the activities.

4.2.1 Sitting

Fig. 9 shows a plot of the acceleration signal for sitting. The signal demonstrates that the x, y, and z values are almost constant, and the z values have larger accelerations than any other values. The statistics of each axis are as follows:

$$\begin{gathered}\mu_x = -1.04, \sigma_x = 0.28 \\ \mu_y = 1.89, \sigma_y = 0.21 \\ \mu_z = 9.5, \sigma_z = 0.08\end{gathered} \tag{8}$$

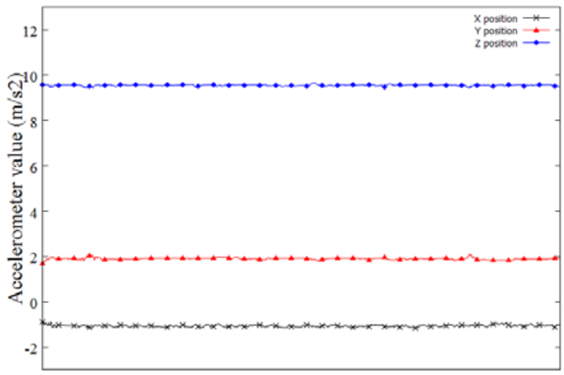

Fig. 9. Acceleration signal for sitting

4.2.2 Standing

Fig. 10 shows an acceleration signal for standing. Similar to sitting, the x, y, and z values are almost constant. Since the axes of the accelerometer are changed, the y values, instead of the z values, have larger accelerations than any other values. The statistics of each axis are as follows:

$$\begin{gathered}\mu_x = -1.79, \sigma_x = 0.14 \\ \mu_y = 9.55, \sigma_y = 0.04 \\ \mu_z = -0.21, \sigma_z = 0.24\end{gathered} \tag{9}$$

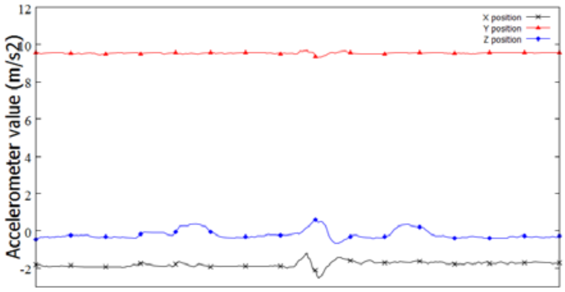

Fig. 10. Acceleration signal for standing

4.2.3 Walking

Fig. 11 shows a graph of the acceleration signal for walking. While the values of sitting and standing are very stable, the values of walking are relatively varied. The y values have larger accelerations than any other values. The statistics of each axis are as follows:

$$\begin{gathered} \mu_x = -1.45, \sigma_x = 0.82 \\ \mu_y = 9.52, \sigma_y = 1.35 \\ \mu_z = 0.82, \sigma_z = 1.18 \end{gathered} \tag{10}$$

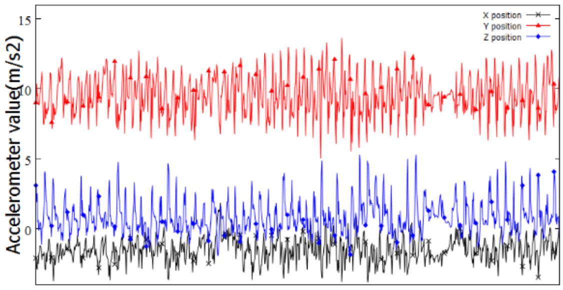

Fig. 11. Acceleration signal for walking

4.3 Feature extraction

According to the properties of each activity and previous work [27-28], we chose the features to be extracted. For a window of accelerations $w = \left((x_1, y_1, z_1), \cdots, (x_n, y_n, z_n)\right)$ where x_i, y_i, and z_i are the i-th acceleration of x, y, and z directions, the average and standard deviation of accelerations are used. Three average (μ_x, μ_y, μ_z) and three standard deviations $(\sigma_x, \sigma_y, \sigma_z)$ are computed as follows

$$\mu_x = \frac{\sum_{i=1}^{n} x_i}{n}, \mu_y = \frac{\sum_{i=1}^{n} y_i}{n}, \mu_z = \frac{\sum_{i=1}^{n} z_i}{n} \tag{11}$$

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu_x)^2}{n}}, \sigma_y = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \mu_y\right)^2}{n}}, \sigma_z = \sqrt{\frac{\sum_{i=1}^{n}(z_i - \mu_z)^2}{n}} \tag{12}$$

In addition, we used three features that consisted of the energy of a signal that the Discrete Fourier Transform generates from the original signal. If X_j is the Discrete Fourier Transform of x_i, the energy of each direction is computed as follows:

$$E_x = \frac{\sum_{j=1}^{n} |X_j|^2}{n}, E_y = \frac{\sum_{j=1}^{n} |Y_j|^2}{n}, E_z = \frac{\sum_{j=1}^{n} |Z_j|^2}{n} \tag{13}$$

Thus, nine features, consisting of the average, standard deviation, and energy of three directions, are used for PSO using ABC. Table 1 shows an example of the feature vectors for each activity.

Table 1. Example of feature extraction

	Sitting	Standing	Walking
μ_x	-0.142	-2.281	-1.656
μ_y	0.149	9.347	9.597
μ_z	10.113	-0.600	0.531
σ_x	0.005	0.008	0.188
σ_y	0.022	0.009	0.602
σ_z	0.000318	0.006	0.441
$Energy_x$	0.16355	41.64	22.208
$Energy_y$	0.30679	698.951	739.81
$Energy_z$	818.296	2.886	3.6265

5. Experiments

In this section, we evaluate the activity recognition algorithm using the ABC technique, and compare our approach to PSO without ABC, as well as other algorithms such as MLP [8], C4.5 [9], and SVM [10]. To alleviate the computational complexity of PSO, we adopted intelligent dynamic swarm [29] that was shown to be 30% faster than traditional PSO. For training dataset, we intentionally collected the accelerations from a triaxial accelerometer, which is embedded in the smartphone during three activities, each of which was performed for five minutes, at a sampling rate of 16 Hz. For test dataset, we collected accelerations in real life for two hours, with sampling periods ($=$ (sampling rate)$^{-1}$) of 2^{-4}, 2^{-2}, 1, 4, 8, 16, and 32 seconds. To compute the accuracy, the intervals of accelerations were manually labeled as one of the three activities. We used an Android-powered smartphone with an application implemented to collect accelerations, as shown in **Fig. 12**.

5.1 Evaluation for sitting

Fig. 13 shows the experiment results for sitting. Since the acceleration signal is very stable for sitting, the experiment shows that all methods recognize sitting with high accuracy. When the sampling period is shorter than or equal to 4 seconds, the accuracy of each algorithm is larger than 98%. If a sampling period increases, then the accuracies drop significantly. For instance, PSO without ABC, MLP, C4.5, and SVM show 97.7%, 94.8%, 94.0%, and 96.8% accuracies at the sampling period of 16 seconds, and 92.6%, 89.4%, 87.6%, and 92.9% accuracies at 32 seconds. PSO with ABC, however, shows the best performance; its accuracy is 98.0% even at a sampling period of 32 seconds.

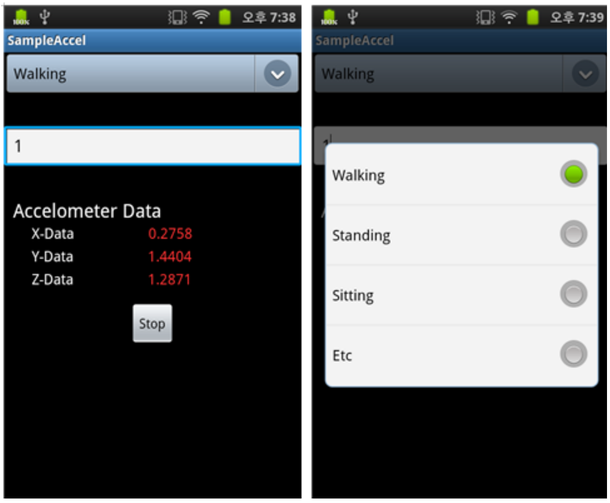

Fig. 12. Application for data collection

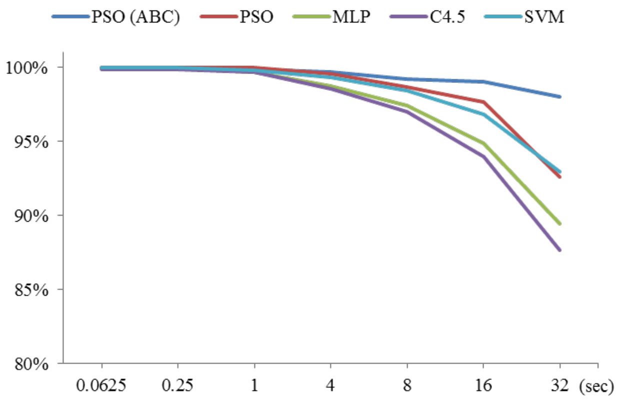

Fig. 13. Experiment results for sitting

5.2 Evaluation for standing

Fig. 14 shows the experiment results for standing. Since the number of accelerations for standing is much smaller than that for sitting, the results looks unstable. Still it verifies that PSO with ABC is better than the other methods. At the sampling period of 4 seconds, most of algorithms are comparable in accuracy. When the sampling period increases, the accuracy of PSO with ABC slightly increases at a sampling period of 8 seconds while the accuracy of PSO decreases. Especially, PSO with ABC and PSO without ABC show 80.9% and 79.2% accuracy at the sampling period of 4 seconds, respectively. When the sampling period is 8 seconds, PSO with ABC shows a robust accuracy of 83.6% while PSO without ABC shows a declined accuracy of 74.1%. This is because ABC catches unclassified data that are generated due to lower sampling rates. Therefore, this experiment shows the effectiveness of the ABC approach.

5.3 Evaluation for walking

Fig. 15 shows the experiment results for walking. The walking results are relatively unstable because walking is a dynamic activity. Nonetheless, the accuracies of all the algorithms are

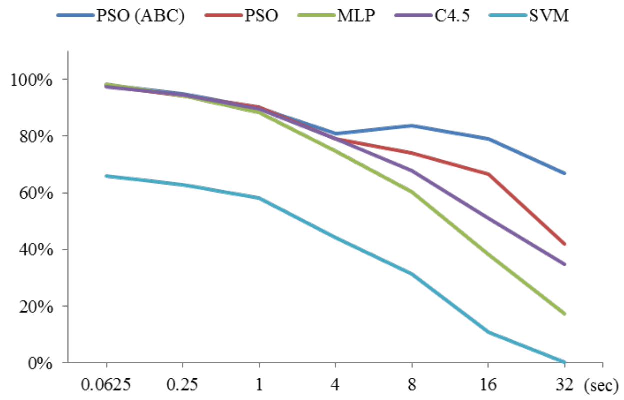

Fig. 14. Experiment results for standing

larger than 85% when a sampling period is smaller than or equal to one second. If a sampling period increases, then the accuracies drop significantly. For instance, PSO without ABC, MLP, C4.5, and SVM show 89.2%, 98%, 97.2%, and 96.4% accuracies, respectively, when the sampling period is one second. The accuracies, however, become 69.8%, 90.9%, 87.9%, and 72.7% at the sampling period of 8 seconds, and become 50.0%, 66.7%, 77.8%, and 44.4% at 32 seconds. The accuracy of PSO with ABC, however, is less reduced than that of the others. The accuracy is 89.2% at one second, 86.5% at 8 seconds, and 85.0% at 32 seconds. Especially PSO with ABC outperforms the other algorithms when a sampling period is 32 seconds; its accuracy is at least 7% better than the others'.

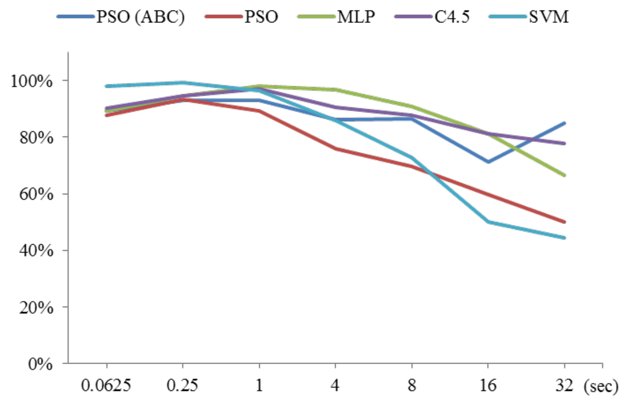

Fig. 15. Experiment results for walking

5.4 Discussion

The experimental results show that the accuracy of the PSO with the ABC approach is better than that of PSO without ABC and the others when a sampling period is relatively large. The ABC approach improves the overall performance by dealing with the outliers that are not in the boundary of the class. For instance, PSO with ABC shows 6%, 25%, and 35% better results than PSO without ABC for sitting, standing, and walking, respectively, with a sampling period of 32 seconds. Especially PSO with ABC is the only algorithm that guarantees at least 80% accuracy for every activity at a sampling period of smaller than or equal to 8 seconds.

Table 2 shows the average activity recognition results of all sampling periods for each algorithm with respect to each activity. Since sitting data are relatively simple, every algorithm accurately identify "sitting" for every sample period. For standing, PSO with ABC outperforms the others because PSO with ABC recognizes "standing" consistently even when a sampling period is large. For walking, PSO with ABC is comparable to MLP and C4.5 and outperforms PSO without ABC and SVM.

Table 2. Overall average activity recognition results for each algorithm

	Sitting	Standing	Walking
PSO (ABC)	99.39%	84.84%	86.34%
PSO	98.35%	77.72%	75.11%
MLP	97.15%	67.39%	88.24%
C4.5	96.65%	73.50%	88.52%
SVM	98.18%	38.95%	78.09%

Table 3 shows the average accuracy differences between the successive sampling periods for each algorithm. The values indicate the tolerance to changes in the sampling periods without a steep decrease in the activity recognition accuracy. The results show that PSO with ABC is the most tolerant algorithm that can recognize activities without losing too much accuracy.

Table 3. Average accuracy differences between the successive periods for each algorithm

	Sitting	Standing	Walking
PSO (ABC)	0.28%	4.48%	0.61%
PSO	1.06%	7.97%	5.42%
MLP	1.51%	11.55%	3.23%
C4.5	1.75%	8.97%	1.80%
SVM	1.01%	9.40%	7.64%

6. Conclusions

This paper proposed PSO with ABC to solve the difficulty of human activity recognition due to the fact that the accuracies of activity recognition algorithms, such as MLP, C4.5, and SVM, become lower as a sampling rate decreases. Although a high sampling rate is neccesary for high accuracy, it is not desirable for actual smartphone usage, battery consumption, or storage occupancy. To maintain recognition accuracies, we started from PSO, which is a relatively consistent algorithm in sampling rates, and we proposed PSO with an ABC approach. ABC is tolerant of small sampling rates in that it identifies all data by adjusting the classification boundary of each activity. We evaluated PSO with ABC and other algorithms with lifelog data from a triaxial accelerometer in a smartphone, with sampling periods of 2^{-4}, 2^{-2}, 1, 4, 8, 16, and 32 seconds. The experimental results show that PSO with ABC has better tolerance to declines of sampling rates than PSO without ABC and other classification methods. In particular, PSO with ABC is 6%, 25%, and 35% better than PSO without ABC for sitting, standing, and walking, respectively, with a sampling period of 32 seconds. PSO with ABC is the only algorithm that guarantees at least 80% accuracy for every activity at a sampling period of smaller than or equal to 8 seconds.

As future work, we are planning to examine our method to detect more activities, such as walking up and down, or running, as well as sitting, standing, and walking, while preserving a high precision. We are also considering a way of coping with exceptional cases, for instance, calling or charging, or keeping the device in places other than a pocket, while recording the lifelog data.

References

[1] T. Ito, T. Ishihara, Y. Nakamura, S. Muto, M. Abe, and Y. Takagi, "Prospects for Using Lifelogs in the Medical Field," *NTT Technical Review*, vol. 9, no. 1, pp. 1-5. 2011. https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201101fa7.html

[2] J. Heo, M. Chun, K. Lee, Y. Oh, O. Noh, and R. Park, "Effects of a Smartphone Application on Breast Self-Examination: A Feasibility Study," *Healthcare Informatics Research*, vol. 19, no. 4, pp. 250-260, 2013. Article (CrossRef Link).

[3] Y. Kwon, K. Kang, C. Bae, H. Chung, and J. Han, "Lifelog Agent for Human Activity Pattern Analysis on Health Avatar Platform," *Healthcare Informatics Research*, vol. 20, no. 1, pp. 69-75, 2014. Article (CrossRef Link).

[4] J. Kim, "Health Avatar: An Informatics Platform for Personal and Private Big Data," *Healthcare Informatics Research*, vol. 20, no. 1, pp. 1-2, 2014. Article (CrossRef Link).

[5] S. Heo, K. Kang, and C. Bae, "Lifelog Collection Using a Smartphone for Medical History Form," *Lecture Notes in Electrical Engineering*, vol. 108, pp. 575-581, 2011. Article (CrossRef Link).

[6] K. Kang, S. Heo, C. Bae, and D. Han, "Mobile Health Screening Form Based on Personal Lifelogs and Health Records," *Lecture Notes in Electrical Engineering*, vol. 108, pp. 557-564, 2011. Article (CrossRef Link).

[7] D.G.M. Zwartjes, T. Heida, J.P.P. van Vugt, J.A.G. Geelen, and P.H. Veltink, "Ambulatory Monitoring of Activities and Motor Symptoms in Parkinson's Disease," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2778-2786, 2010. Article (CrossRef Link).

[8] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, MIT Press, 1986, pp. 318-362, 1986. http://dl.acm.org/citation.cfm?id=104279.104293

[9] J.R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers Inc., San Francisco, 1993. http://dl.acm.org/citation.cfm?id=152181

[10] C. Cortes and V. Vapnik, "Support-Vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995. Article (CrossRef Link).

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. of IEEE International Conference on Neural Networks (ICNN)*, pp. 1942-1948, 1995. Article (CrossRef Link).

[12] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," in *Proc. of IEEE International Conference on Evolutionary Computation (ICEC)*, pp. 69-73, 1998. Article (CrossRef Link).

[13] C. Bae, W.-C. Yeh, N. Wahid, Y.Y. Chung, and Y. Liu, "A New Simplified Swarm Optimization (SSO) Using Exchange Local Search Scheme," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 6, pp. 4391-4406, 2012. http://www.ijicic.org/ijicic-11-06072.pdf

[14] D. Lee and S. Lee, "Vision-Based Finger Action Recognition by Angle Detection and Contour Analysis," *ETRI Journal*, vol. 33, no. 3, 2011. Article (CrossRef Link).

[15] J. Chang and S. Nam, "Fast Random-Forest-Based Human Pose Estimation Using a Multi-scale and Cascade Approach," *ETRI Journal*, vol. 35, no. 6, 2013. Article (CrossRef Link).

[16] Z. Gao, H. Zhang, A.-A. Liu, Y.-B. Xue, and G.-P. Xu, "Human Action Recognition Using Pyramid Histograms of Oriented Gradients and Collaborative Multi-task Learning," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 2, 2014. Article (CrossRef Link).

[17] R. Jafari, W. Li, R. Bajcsy, S. Glaser, and S. Sastry , "Physical Activity Monitoring for Assisted Living at Home," in *Proc. of Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 213-219, 2007. Article (CrossRef Link).

[18] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity Classification Using Realistic Data from Wearable Sensors," *IEEE Transactions on Information Tech-*

nology in Biomedicine, vol. 10, no. 1, pp. 119-128, 2006. Article (CrossRef Link).

[19] S.L. Lau, I. Konig, K. David, B. Parandian, C. Carius-Düssel, and M. Schultz, "Supporting Patient Monitoring Using Activity Recognition with a Smartphone," in *Proc. of Int. Symposium on Wireless Communication Systems (ISWCS)*, pp. 810-814, 2010. Article (CrossRef Link).

[20] M. Lee, A.M. Khan, and T. Kim, "A Single Tri-axial Accelerometer-based Real-time Personal Life Log System Capable of Human Activity Recognition and Exercise Information Generation," *Personal and Ubiquitous Computing*, vol. 15, no. 8, pp. 887-898, 2011. Article (CrossRef Link).

[21] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele, "ADL Recognition based on the Combination of RFID and Accelerometer Sensing," in *Proc. of Int. Conf. on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pp. 258-263, 2008. Article (CrossRef Link).

[22] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, "Activity Recognition from Accelerometer Data," in *Proc. of American Association for Artificial Intelligence (IAAI)*, pp. 1541-1546, 2005. http://dl.acm.org/citation.cfm?id=1620107

[23] S.L. Lau and K. David, "Movement Recognition Using the Accelerometer in Smartphones," *Future Network and Mobile Summit*, pp. 1-9, 2010.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5722356

[24] M.Z. Uddin, N.D. Thang, J. Kim, and T. Kim, "Human Activity Recognition Using Body Joint-Angle Features and Hidden Markov Model," *ETRI Journal*, vol. 33, no. 4, pp. 569-579, 2011. Article (CrossRef Link).

[25] S. Heo, K. Kang, and C. Bae, "Activity Classification Using a Single Tri-axial Accelerometer of Smartphone," *Lecture Notes in Electrical Engineering*, vol. 182, pp. 269-275, 2012.
Article (CrossRef Link).

[26] L. Bao, "Physical Activity Recognition from Acceleration Data under Semi-Naturalistic Conditions," *Master's Thesis*, Massachusetts Institute of Technology, 2003.
http://dspace.mit.edu/handle/1721.1/87432

[27] U. Maurer, A. Smailagic, D.P. Siewiorek, and M. Deisher, "Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions," in *Proc. of the Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 113-116, 2006. Article (CrossRef Link).

[28] J.-Y. Yang, Y.-P. Chen, G.-Y. Lee, S.-N. Liou, and J.-S. Wang, "Activity Recognition Using One Triaxial Accelerometer: A Neuro-fuzzy Classifier with Feature Reduction," in *Proc. of Int. Conf. on Entertainment Computing (ICEC)*, pp. 395-400, 2007. Article (CrossRef Link).

[29] C. Bae, W. –C. Yeh, Y. Y. Chung, and S. –L. Liu, "Feature Selection with Intelligent Dynamic Swarm and Rough Set," *Expert Systems with Applications*, vol. 37, no. 10, pp. 7026-7032, 2010. Article (CrossRef Link).

Yongjin Kwon received his BS in computer science from Pohang University of Science and Technology (POSTECH), Pohang, Rep. of Korea in 2009, and his MS in computer science from Seoul National University, Seoul, Rep. of Korea in 2012. Since 2012, he has been working with Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. His research interests include database systems, lifelog data mining, and information retrieval.

Seonguk Heo received his BS in computer science from Korea University of Technology and Education, Cheonan, Rep. of Korea in 2011, and his MS in computer science from University of Science and Technology (UST), Daejeon, Rep. of Korea in 2013. Since 2013, he has been working with Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Rep. of Korea. His research interests include lifelog data mining, personal computing, and next generation computing.

Kyuchang Kang received his PhD in computer science from Chungnam National University, Daejeon, Rep. of Korea in 2009. Since 2001, he has been working with Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea, as a principal researcher. His research interests include open software platform, lifelog data mining, personal computing, and next generation computing.

Changseok Bae received his PhD in electronics engineering from Yonsei University, Seoul, Rep. of Korea in 2003. Since 1989, he has been working with Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea, as a principal researcher. He is currently the leader of the Human Computing Section. His research interests include lifelog data mining, personal computing, and next generation computing.