Improving Fault Tolerance of Wearable Sensor-based Activity Recognition Techniques

Ryoma Uchida
Toyohashi University of Technology
hibarigaoka 1-1, tenpakucho, toyohashi-shi, aichi-ken, JAPAN
ryoma@usl.cs.tut.ac.jp

Hiroto Horino
C-Tech, corp.
horino@usl.cs.tut.ac.jp

Ren Ohmura
Toyohashi University of Technology
hibarigaoka 1-1, tenpakucho, toyohashi-shi, aichi-ken, JAPAN
ren@tut.jp

http://dx.doi.org/10.1145/2494091.2495984

Abstract

Existing wearable sensor-based activity recognition techniques lack fault tolerance in the case of sensors data loss, such as communication disconnection and sensor failure. Compensating for missing data is one method to improve robustness and can be done by three levels in activity recognition: raw data level, feature value level, and classifier level. Our study proposes a method to compensate for the missing sensor data using an ARAR algorithm and compares this method with a previous method for compensating for the feature value using kernel regression in the feature value level. The ARAR algorithm method predicts future data from existing sequence data. We conducted some experiments to verify the usefulness of the proposed methods. Specifically, the prediction performance was evaluated by applying the ARAR algorithm to compensate for one to five successive windows. As a result of our test data, the F-measure rate was 73.4% in the case of sensor data loss. The ARAR algorithm compensation for one and two successive windows increased the F-measure to 76.8%. Overall, the ARAR algorithm method effectively compensates for instantaneous communication disconnection. On the other hand, the kernel regression method is especially compensates for burst communication disconnection.

Therefore, we need to change the compensation method depending on sensor error patterns. Thus, we improved robustness of the activity recognition system by compensating for sensor data loss.

Author Keywords

Activity recognition, improving fault tolerance, completing defective value

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms

Algorithm, Experimentation, Performance

Introduction

Activity recognition techniques with wearable sensors are some of the most important technologies to build a context aware system [11]. These techniques can be used in diverse cases, for example, nursing [3], personal life logs [10], and air plane mechanics [1]. If the system is achieved, we can support usual or unusual activity of users and make them more comfortable by automatic control [7], [8].

Normally, an activity recognition system collects some data from a variety of wireless sensors (*ex.* GPS, acceleration sensor, and angular velocity sensor). The system transforms these raw data into feature values at every window function. Especially notable feature values are maximum, minimum, mean, variance values, and power spectrum. Most current activity recognition techniques are based on machine learning algorithms [9]. They basically compare feature values extracted from sensor data between training time and recognition time. Then the system recognizes user's activity information. In general, using multiple wearable sensors can increase the accuracy of recognizing complex activities.

However, existing wearable sensor-based activity recognition techniques lack fault tolerance in the case of sensor data loss, such as communication disconnection and sensor failure. Data loss from even only one sensor causes the loss of some feature values and can degrade the recognition accuracy. Moreover, the algorithm cannot give any output with incomplete sensor data (feature value) in the worst case. In our project for monitoring nursing activities, the activities can be monitored for only about 60% of the whole experiment time due to communication failure of Blue-tooth between the body-worn sensor and wearable storage. That is, about 40% of activity data must be compensated for. If nurses engage in medical malpractice when the system connection error occurs, we cannot alert them to stop them making medical errors.

Data loss patterns brought about in two ways. One is instantaneous communication disconnection. This error can happen any time we use wireless devices. The other is burst error. This error can be caused by sensor failure. Both cases stop the existing activity recognition systems, so we need to solve these problems.

While coping with missing data in classification has been studied in other fields, such as voice recognition [5], there are few studies on activity recognition. Because of this, voice recognition systems are the most cutting-edge, robust techniques. They compensate for missing data and have high recognition accuracy. Missing data can be compensated for by three levels in activity recognition: raw data level, feature value level, and classifier level. Hesam *et al.* proposed a classifier level compensation method [4]. However their approach requires matrix calculation including obtaining inverse matrix to find a compensating value, and it needs relatively high computational power. Our previous study proposed a feature value level compensation method using multiple sensors [13]. We proposed two methods. One, compensated for missing feature values by linear regression (multiple regression). The other does so by non-liner regression (kernel regression). The results showed the kernel regression method outperforms multiple regression. Murao *et al.* proposed a raw data level compensation method [6]. Their approach compensated for missing raw data by a fixed value. A fixed value is stored in their database. However, this approach cannot cope with unknown values.

Therefore, we proposed a technique for coping with the problem. Our study tried to compensate for the missing sensor data using an ARAR algorithm in the raw data level and compared this method with a previously proposed method for compensating for the feature value using kernel regression in the feature value level. Then, we clarify each method's effectiveness. The ARAR algorithm predicts future data from existing sequence data by applying memory shortening transformation and fitting an AR model to the transformed data. This method is known to predict many sequence data accurately. In our proposed method, an ARAR model is created from sensor data obtained when data are lost. The kernel regression method depends on only calculated feature values and does not restrict feature calculation and classification algorithm.

Also, we compare and combine methods. As a result, we can improve performance by choosing a different method for every activity or sensor position. Thus, our methods can improve fault tolerance by combining various methods.

Problems for activity recognition system

Activity recognition techniques usually involve some wearable sensors, for example, an acceleration sensor or angular velocity sensor. Recognition systems can receive multiple sensor data from each wearable sensor. When the system has collected all sensor data, we extract features of the data. Therefore, the raw data are transformed into the feature values such as the mean, maximum, minimum, and variance values. An activity recognition system treats these feature values as a set of feature vectors. After the feature extraction, the system uses a matching algorithm between the feature vector and training data set. After the system has executed the matching algorithm, user can obtain the results of activity recognition.

However, existing wearable sensor-based activity recognition techniques lack fault tolerance in the case of sensor data loss, such as communication disconnection, battery death, and sensor failure. Data loss from even only one sensor causes the loss of some feature values and can degrade the recognition accuracy. Moreover, the algorithm cannot give any output with incomplete sensor data (feature value) in the worst case. Specifically, for these reasons, compatibility between the extracted feature vector and training data set is lost. Also, the matching algorithm cannot recognize activities. Therefore, we need to maintain dimension number compatibility between the feature vector and training data.

Also, we have to consider sensor error patterns, which come in two types. One is instantaneous communication disconnection. This error can happen any time because of a barricade between a data signal receiver and wearable sensors if we use wireless devices. This error causes short-time data loss, but the disconnected sensor can be reconnected. The other is burst error. This error can be caused by battery death or sensor failure. This error causes long-time data loss, and the disconnected sensor cannot be reconnected. In any event, these cases stop existing activity recognition systems, and users cannot obtain results of activity recognition. Therefore, we need to solve these problems.

Compensating for missing data is one method to solve these problems and can be done by three levels in an activity recognition system.

Raw data level compensates for acceleration data or gyro data missing from sensor data. This study focuses on this level because of a related work [6] is not considered about unexpectedly sensor error. They decided to use a part of multiple sensor system. But if the system lost a part of sensor's data by communication disconnection, their method cannot support it. This lack of consideration of this level fault tolerance approach.

Feature value level compensates for mean values and variance values missing from feature values. Our previous study proposed a feature value level method using kernel regression [13]. This method revealed for improving fault tolerance of activity recognition system because of compensating for missing feature values. So they are already enough to consideration of feature value level approach.

Classifier level compensates for missing recognition results. The easiest method is to decrease the feature vector's dimension number in a training data set. The decreased feature vector is the same as the missing feature value.

Method

We proposed a technique for coping with data loss. In this study, when raw data were lost between body-worn sensor and a wearable storage, we compensated for the missing raw data by using past sequence data. Our study tried to compensate for the missing sensor data using an ARAR algorithm in raw data level. Specifically, we create AR regression models from the relationships in past time series data. Then missing data are complemented for by an AR regression model. Also, we compare this method with our previously proposed one. The previous approach compensates for missing feature values by using a kernel regression model. Thus, we compare the ARAR algorithm method in the raw data level and the kernel regression method in the feature value level to clarify their effectiveness.

ARAR algorithm

The ARAR algorithm predicts future data from existing sequence data by applying memory shortening transformation and fitting the AR model to the transformed data. It is known to predict many sequence data accurately [12].

First, the ARAR algorithm ascertains whether the past time series data's self correlation is divergence or convergence. If it is divergence, the data are transformed by the memory shortening transformation in Figure 1. β is calculated by Eq. 1 each case of the $\tau = 1, 2, \ldots, 15$. Then, we choice of the β which make Eq. 1 into the smallest value. In case of the $\tau > 2$, we use Eq. 2 , and otherwise use Eq. 3. Right-hand y_t is normal past sensor data.

$$\min \frac{\sum_{t=\tau+1}^{n} [Y_t - \beta Y_{t-\tau}]^2}{\sum_{t=\tau+1}^{n} Y_t^2} \tag{1}$$

$$\hat{y}_t = y_t - \beta_\tau y_{t-\tau} \tag{2}$$

$$\hat{y}_t = y_t - \beta_1 y_{t-1} - \beta_2 y_{t-2} \tag{3}$$

After the transformation, we again check whether the time series data's self correlation is divergence or convergence. If it is convergence, we create the AR model, but if it is divergence, we continue memory shortening transformation until convergence is achieved.

Next, we introduce fitting the AR model. The AR model is shown in Eq. 4. y_t is an objective variable, y_{t-n} is an explanatory variable, and β is an autoregressive coefficient.

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_n y_{t-n} + \varepsilon \tag{4}$$

β_n is calculated by Yule-Walker's equation, which is Eq. 5. ρ is the autocorrelation function.

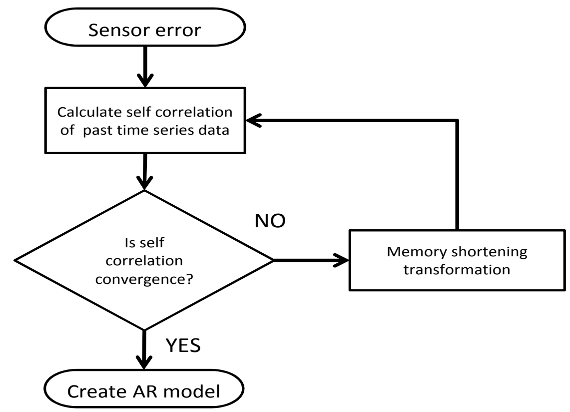

Figure 1: Memory shortening transformation.

$$\begin{bmatrix} 1 & \rho(1) & \ldots & \rho(n-1) \\ \rho(1) & 1 & \ldots & \rho(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(n-1) & \rho(n-2) & \ldots & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(n) \end{bmatrix} \tag{5}$$

In this study, we predict lost sensor data by using Eq. 4. y_t is predicted lost sensor data (raw data), and y_{t-n} is the available value that is past time series data.

Kernel Regression

Our previous study proposed kernel regression method in the feature value level. Kernel regression creates a model given by

$$y_i = \sum_{i=1}^{n} \beta_i K(x_i, x) + \beta_0 \tag{6}$$

Here, K is a kernel function that maps data space to high dimension spaces. We use a polynomial kernel for feature extraction. Linear model fitted to data in the high dimension space become non-linear fitting in the original data space. That is, regression parameters β can be decided in a linear method while a complex non-linear model is obtained. There are some variations in kernel regressions, and we used SMOreg in Weka [14], which is an implementation of a support vector machine for regression. Variable y_t corresponds to a missing feature value. x is an explanatory variable the data of which can normally obtain sensor data's feature values.

EXPERIMENTS

We conducted some experiments to verify the usefulness of the proposed methods. The ARAR algorithm method compensates for missing raw data by using past time series sensor data gathered before disconnection. Also, the kernel regression method compensates for missing feature values.

Experiment for HASC2012corpus

First we need to monitor trends in the ARAR algorithm method. Therefore, we experimented using HASC2012corpus data which include 41 person data (64 accelerometer's data) and 6 activities (stand, walk, skip, jog, upstairs, and downstairs). HASC2012corpus data use only one sensor. We tried to compensate for each data's one to five successive windows using the ARAR algorithm method.

Experiment for our test data

Next we evaluated our method, how different work for each sensors position or user's each activities. So we created our test data using multiple sensor because HASC2012corpus use only one sensor. We conducted experiments using six tri-axial accelerometers and gyro WAA-006 [2], which could measure ± 3G at 100Hz, with five subjects, who were university students. Each subject was asked to put the accelerometers on both his/her wrists, both ankles, chest (in a pocket), and lower back and to "clap hands", "fold arms", "sit", "run", "skip", "stand" and "walk" during thirty minutes without any extra instructions from the experimenter.

Experimental methodology

Feature values were mean and standard deviations. These values were computed from each axis of both accelerometers and gyro and were extracted from a sliding window of 2.56 seconds width shifting by 1.28 seconds. Training data were 30 minutes of sequence data (including seven activities). A support vector machine (SMO in Weka[14]) was used as the classification algorithm. The prediction performance was evaluated by applying the ARAR compensation to one to five successive windows or using a kernel regression model. Also, we use data from one window (256 samples) for past time series data.

Here, we compared the following four conditions.

Normal situation (hereafter referred to as "normal") means no data loss, and six sets of accelerometer and gyro data are always available. This can be thought to obtain the maximum recognition rate.

Decreased sensor classifier (hereinafter "de-clas") gives the situation in which training data were created by removing one set of sensor data, and recognition is done by the same sensors. That is, five sensors are used in training and recognition, and six cases (right wrist, left wrist, right ankle, left ankle, waist, and chest) are averaged to obtain the final result.

Compensating by ARAR algorithm (hereinafter "ARAR") gives the situation in which one set of sensor data is lost and the missing sensor data are compensated for by the ARAR algorithm. ARAR algorithm is able to generate an autoregressive model, which is known to predict many sequence data accurately. One sensor has three axes of both accelerometer data and gyro data, so we need to compensate for 12 sets of missing data. The result was averaged from six cases.

Compensating by kernel regression (hereinafter "kern-reg") gives the situation in which one set of sensor data is lost and the missing feature value is compensated for by kernel regression. Kernel regression is able to generate a non-linear regression model in higher-dimensional space, so this regression model is higher accuracy than a linear model. One sensor has three axes of both accelerometers and gyro, and three feature values are extracted from data of each axis. Thus, 12 missing features are calculated from 60 available values. The result was averaged from six cases.

Results

F-measure values are calculated by $\frac{2*Precision*Recall}{Precision+Recall}$.

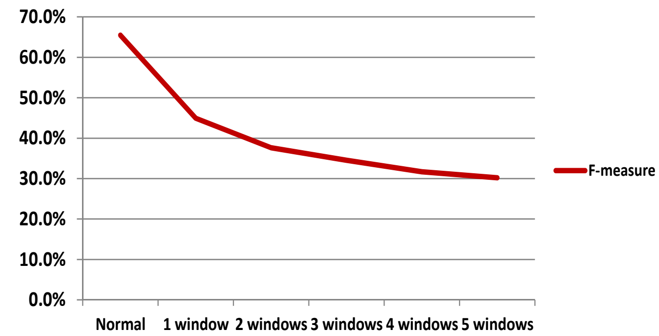

Figure 2: Results of HASC2012corpus using the ARAR algorithm mehtod. Normal is no compensate for.

We compared each method's performance from one to five windows. HASC2012corpus results and our data results are shown in Figure 2 and 3.

Result of HASC2012corpus

We tried to check of the ARAR algorithm method performance using HASC2012corpus. Figure 2 showed that performance gradually reduced from one window case to five window case. Normal situation is 65.5% accuracy of F-measure. In case of the compensate for one window data, F-measure performed 44.9% and five windows case performed 30.2% accuracy. It means the ARAR algorithm method can compensate for missing data in raw data level but also difficult of compensate for missing data for a long time. For example, in case of the two window case, first ARAR algorithm predict one window's missing data from past normal time series data. Next ARAR algorithm predict second window's missing data using predicted one window missing data by ARAR algorithm. So, repeating prediction using predicted data is very difficult.

Result of our data

In the results of Figure 3, the ARAR algorithm outperforms the decrease sensor classifier method when compensating for one and two window's worth of raw data. For normal, the F-measure rate is 78.9% but decreases 5.5% in the case of sensor data loss. The ARAR compensation for one and two windows increased the F-measure rate of 76.8%. Also, kernel regression increased the F-measure rate to 74.1%. The ARAR algorithm can improve fault tolerance over a short time, meaning the range of one or two windows. In contrast, the kernel regression method can improve fault tolerance over a long time, meaning more than three window's worth of data loss. However, the ARAR algorithm accuracy decreased in the case of long-time data loss. That is to say, the ARAR algorithm has difficulty continuing to compensate for missing data for a long time. In the experiment, we use one window's worth of data for past time series data, so one window's worth of data can compensate for two shifting window's worth of data.

Figure 4 shows recognition accuracy of all methods for each activity. One window was used in the case of the ARAR algorithm method. In the case of clapping hands, the decrease sensor classifier method performs the worst, while the ARAR algorithm method and kernel regression method perform better. In the case of folding arms, performance results are similar to those of clapping hands. In the case of running, the kernel regression method performs the worst, but the ARAR algorithm performs the best. The result of sitting, skipping, and standing cases do not differ much between methods. In the case of walking, kernel regression method performs the worst. Clapping hands and folding arms depend on arm behavior and short time scale, which means that the ARAR algorithm

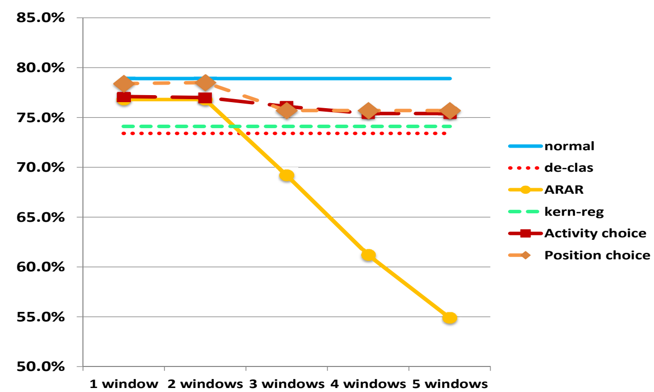

Figure 3: Results of all. Activity choice is changing method for each activity. Position choice is changing method for each sensor position.

method has an advantage for arm activity and short cycle activity. Also, the kernel method has an advantage for long time scales. There is almost no variation in the standing activity results since if we stand in the same place, our body acceleration is unchanged, so standing is easy to recognize regardless of the number of sensors. Also, the ARAR algorithm can perform well in short cycle activities like running, so its running performance was improved, but its walking performance was not because it has a longer cycle than running. Moreover, sitting is not cycle activity for each positions. If waist sensor connection fault, it is difficult to recognize sitting or standing. So, our method can not good compensate for sitting activity's missing raw data and feature values.

Overall, we can improve fault tolerance in a short time by using the ARAR algorithm. Also, our methods have an advantage for arm activities like clapping hands and folding arms. In the decreased sensor method, clapping

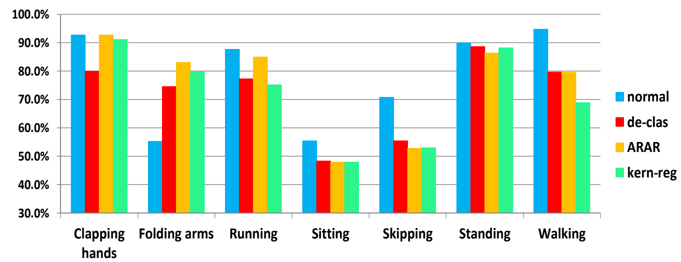

Figure 4: Results of all activities. ARAR algorithm compensates for one window's worth of data.

hands accuracy is 79.9% and folding arms accuracy is 74.5%. In the ARAR algorithm, clapping hands accuracy is 92.7% and folding arms accuracy is 83.0%. Therefore, the ARAR algorithm can improve accuracy about 10.5% for these activities. Our trivial movements like typing on a keyboard, reading a book, open the door, etc. depend a lot of on arm behavior. Thus, our proposed methods are suitable for increasing fault tolerance of wearable sensor-based activity recognition systems.

Discussion

We evaluated each position's recognition result accuracy to prove the relationship between sensor positions and recognition accuracy. Each position's F-measure rate is shown from Figure 5 to Figure 7.

The ARAR algorithm method achieved better position accuracy for both wrist than the decrease sensor classifier method. The reason is that the acceleration of such wrist and ankle positions' changes significantly, so the ARAR algorithm's regression model can fit past time series data of wrists and ankles. However, if data loss occurs for a long time, such as more than three windows, the ARAR algorithm's regression model cannot predict accurately. This means it has difficulty continuously predicting. Therefore, both wrists and ankles need to be switched to another method if data loss occurs for a long time. In the case of wrists, we should change to the kernel regression method when we predict from more than three windows' worth of data. Also, in the case of ankle, we should change to the decrease classifier method when we predict from more than three windows' worth of data.

Waist position simulates the condition of a user's pocket for a smart phone. Thus, waist position accuracy using the ARAR algorithm method remains about the same as using the decreased sensor classifier method. On the other hand, the kernel regression method performs well. However, in the future, if we are able to recognize activities using only a smart phone (one sensor), the kernel regression method and decreased classifier method will not be able to be applied for compensation here. Thus, we should use the ARAR algorithm method. If the waist pocket sensor disconnects when in use here, the ARAR algorithm method is fine.

In the case of chest pocket position, the decreased classifier method performs better than normal case because the sensor is inclined by the subject's vigorous action. Thus, the ARAR algorithm method and kernel regression method cannot fit the regression model. We need to remove vigorous activity data. That way, we can create better regression models for compensation.

With all these factors, if sensor data are lost for a short time, the ARAR method is the most effective approach. However, the ARAR method is unsuitable for long-time connection faults, so we need switch to another method from the ARAR method. If the fault case is a wrist, the method should change to the kernel regression method. Also if the fault case is an ankle, the method should change to the decrease classifier method. This way, recognition accuracy can be maintained.

We have to consider sensor error patterns: instantaneous communication disconnection, which causes errors for a short time, and burst error, which causes long-time data loss. ARAR algorithm can be applied to instantaneous communication disconnection but not to burst error. Therefore, the kernel regression method should be applied to burst error. By switching compensation methods to other methods, we can best compensate for missing data and increase fault tolerance.

This time, we experimented using data loss from one sensor. However, multiple wearable sensor-based activity recognition systems potentially have multiple-sensor data loss cases (one-sensor data loss, two-sensor data loss, etc.). Also, the system may potentially suffer all-sensor communication disconnection. If all sensors are disconnected, no matter if the error case is instantaneous or burst, the kernel regression method and decreased sensor classifier method cannot be applied to compensate for missing data. However, the ARAR algorithm method can be applied to compensate for missing data, so the ARAR algorithm is very useful for fault tolerance. And ARAR algorithm very much depends on cyclic of past time series data. So, if past time series data has powerful cycle, ARAR algorithm can fit AR model for past time series data, and high performance of compensate for missing data.

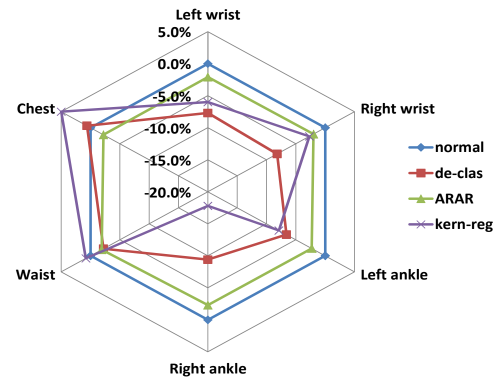

Figure 5: Result of each position in case of one window's data lost.

Here, if we chose methods for each activity (Activity choice) and each sensor position (Position choice), recognition performance maintain high quality as shown in Figure 3. For example, in case of the activity choice, clap is ARAR algorithm method, stand is kernel regression method, skip is decreased classifier method, and in case of the position choice, left wrist is ARAR algorithm method, chest is kernel regression method. Both choices has the better compensation performance of all until the fifth window. This means that we do not have to focus on only one method. Combining various methods is important for fault tolerance, which means we can switch to methods for not only each activity but also each sensor position.

Overall, the ARAR algorithm method effectively compensates for instantaneous communication disconnection. Our proposed method is especially effective in wrist or ankle positions that have large changes in acceleration. However, the ARAR algorithm method cannot predict missing data for a long time. On the other hand, the kernel regression effectively compensates for

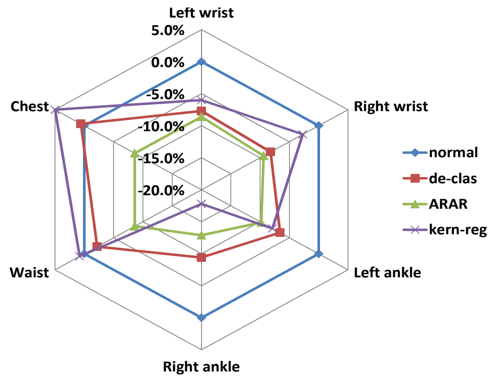

Figure 6: Result of each position in case of three window's data lost.

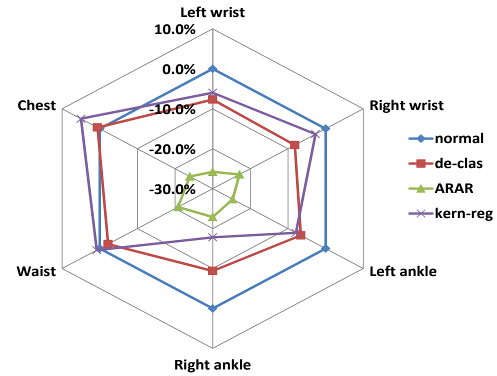

Figure 7: Result of each position in case of five window's data lost.

burst communication disconnection. The kernel regression method can predict missing feature values for a long disconnection time. However, if the user uses only kernel regression method or decreased sensor classifier method, the recognition system cannot treat all sensor data loss cases. Thus, we need to change the compensation method depending on number of disconnected sensors. Also, we can improve performance by choosing a method at every activity or sensor position. As a result, our proposed methods can improve fault tolerance by combining various methods.

Future work & Conclusion

For coping with sensor data loss in accelerometer-based activity recognition, we evaluated an ARAR algorithm method and a kernel regression method for compensating for missing raw data and feature values. The F-measure rate is 73.4% in the case of sensor data loss. The ARAR algorithm compensation for one and two successive window increases the F-measure rate to 76.8%. However, the ARAR algorithm compensation for three to five successive windows is not as good as that of the kernel regression method, so we need to switch to another method in accordance with connection fault time length. As a result, ARAR method is the most effective approach to sensor data loss caused by instantaneous communication disconnection.

On the other hand, the kernel regression method or decreased classifier method is more effective than the ARAR method for sensor data loss caused by burst communication disconnection. However, if a user uses only the kernel regression method or decreased sensor classifier method, the recognition system cannot treat all sensor data loss cases, so we need to change the compensation method depending on the number of disconnection sensors. Also, we can improve the performance by choosing different methods at every activity or sensor position, so our methods can improve fault tolerance by combining various methods. In future work, we will innovate with three compensation methods (raw data level, feature value level, and classifier level) to improve robustness of the activity recognition system.

Acknowledgements

The authors are grateful to five our laboratory members for helping us collect experimental data.

References

[1] Amit Roy Chowdhury, Rama Chelappa.: A Factorization Approach for Activity recognition, Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03.

[2] ATR-Promotions.: WAA-006, http://www.atr-p.com/sensor06 , 2013-5-24.

[3] D Snchez, M Tentori, J Favela.: Activity Recognition for the Smart Hospital, Intelligent Systems, IEEE 2008, pages50-57.

[4] Hesam Sagha, José del R. Millán and Rlcado Chavarrlaga.: A Probablistic Approach to Handle Missing Data for Multi-Sensory Activity Recognition, *UbiComp '10*,OpportunisticUbiquitousSystems-50- , W14(2010).

[5] JK Dixon.: Pattern Recognition with Partly Missing Data, Systems, Man and Cybernetics, IEEE Transactions, Vol. 9, pages 617-621.

[6] Kazuya Murao, Tsutomu Terada, Yoshinari Takegawa, Shojiro Nishio.: A Context-Aware System that Changes Sensor Combibations Considering Evergy Consumption, Pervasive 2008, LNCS 5013, pp. 197-212, 2008.

[7] Licia Capra, Wolfgang Emmerich, Cecilia Mascolo.: CARISMA: Context-Aware Reflective middleware System for Mobile Applications, Software Engineering, IEEE Transactions, pages 929-945.

[8] Matthias Baldauf, Schahram Dustdar, Florian Rosenberg.: A survey on context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007, pages 267-277.

[9] Mitja lusštrek, Boštjan Kaluža.: Fall Detection and Activity Recognition with Machine Learning, Informatica (Slovenia) 33(2), 197-204 (2009).

[10] MW Lee, AM Khan, TS 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, 2011, pages 887-898.

[11] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, Michael L. Littman.: Activity Recognition from Accelerometer Data, In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI'05), Volume 3, Pages 1541-1546.

[12] Peter J. Brockwell, Richard A. Davis.: Introduction to Time series and Forecasting, 2nd edition(Springer, 2002).

[13] Ryoma Uchida, Ren Ohmura.: Preliminary Evaluation of Feature Level Compensation for Missing Data in Multi-sensor Activity Recognition, *UbiComp '12*, Pages 645-646.

[14] University of Waikato.: Weka, http://www.cs.waikato.ac.nz/ml/weka/, 2013-04-11.