Associative Classification for Human Activity Inference on Smart Phones

Yu-Hsiang Peng(✉), Gunarto Sindoro Njoo(✉), Shou-Chun Li, and Wen-Chih Peng(✉)

Department of Computer Science, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan, Republic of China
{eric45670,gunarto.nctu}@gmail.com, wcpeng@cs.nctu.edu.tw

Abstract. With the population of smart phones, the general trend of human activity inference is prospering under a powerful computation capabilities on modern phones. Such an assistant make users life more convenient and help them prevent from unnecessary interferences. In conventional research, the activity inference problem is considered a classification instance, so in this paper we propose an association-based classifier framework (ACF) that aims at exploring the correlation among collected sensor data. Each data consists of multiple sensor readings with a label, e.g., dining, shopping, working, driving, sporting, and entertaining. Note that ACF caters to the discrete data; as a consequence, the continuous sensor readings are needed to be transformed to some discrete groups. Therefore, we propose an Interval Length-Gini Discretization (LGD) method which considers the groups and misclassified cases to obtain the best hypothesis for a given set of data. After an appropriate discretization, we propose one-cut and memory-iteration-based approach to select a set of useful sensor-value pairs for reducing the model size by removing redundant features and guaranteeing an acceptable accuracy. In the experiments our framework has a good performance on real data set collected from 50 participants in eight months, and a smaller size than the existing classifications.

Keywords: Activity recognition · Smart phones · Classification · Associative rule · Discretization · Feature selection

1 Introduction

Human activity recognition is an important issue. Many stream data mining methods have been proposed. Some research in [10] can accurately recognize activities if sensor data is collected from smart environments. Another research in [11] even can predict activity based on video stream data. Other researchers in [2,12] focus on how to recognize simple activities based on video data. But all these approach are not suitable to recognize user activities on smart phones since they need put sensors in stable environments but the sensors on smart phones often face different environments. In recent years, activity recognition

W.-C. Peng et al. (Eds.): PAKDD 2014 Workshops, LNAI 8643, pp. 305–317, 2014.
DOI: 10.1007/978-3-319-13186-3_29

using mobile device is becoming more popular because smart phones has already equipped with various types of sensors, such as: several motion sensors (e.g. accelerometer sensors), environment sensors (e.g. light sensors, and pressure sensors), temporal sensors (e.g. time) and location sensors (e.g. GPS sensors). Previously, many researchers try to use these equipments to recognize user activities. The authors in [13] try to use wearable motion sensors attached in several human body position to recognize motion activities. Other authors in [1,6] use motion sensors on smart phones to detect some simple motion activities, such as walking, running, going up, and going down. Some of them even can roughly detect complex motion activities. These work can achieve high accuracy in mobile device but all of them do not consider the storage issue in smart phones. So, we want to design a framework which can detect user complex life activities and it is suitable on smart phones.

Human activity has lots of types. In this paper we focus on life activities (e.g. shopping, entertainment, sporting, working, transporting and dining). Comparing to action activities (e.g. walking, running, going up and etc.), because life activities can be regarded as the combination of some action activities, life activities are more meaningful for human and detecting them is more challenging. The problem of activity inference can be deemed as classification problem. Logger collect training dataset to build the classifiers. Classifier use the sensor data to recognize user activities. To use classifier on smart phones, there are some limitations. First, recognizing time need to be as short as possible in order to enhance user experience. Then, model size need to be as small as possible so that the model can be built in low layer architecture (e.g. sensor hub [7]). Associative classifier conforms the above two requirements. The detail of associative classifier will be introduced in Sect. 3.1. Associative classifier aims at mining association rules among context information hidden in the training dataset. In general, the classifier composed of association rules has smaller model size than other model. In addition, rule-based classification has another advantage. By using rule-based classification, users can understand their behavior easily by looking at rules in the model. Hence users can observe the model to realize their lifestyle and behavior.

Some sensor data from mobile device are continuous value. So, another issue is how to discretize these data. Discretziation method will influence the performance and model size of classifier. Discretization method has already been a mature research topic. There are two issues in discretization which are how to judge the quality of partition and what is stop condition. For first issue, in this paper, we use two famous discretization method idea. One is Gini index which is proposed by C. W. Gini [5]. Another one is entropy-based discretization standard (e.g. information gain). After these standard had been published, one research compares the performance of these methods [9]. Experiment shows that it is hard to determine which one is better. For second issue, in our observation, having too many slots will cause the classification algorithms to be cumbersome and having too few slots will cause the redundancy and impurity of slots. Therefore, we decide to propose a method called Length-Gini Discretization (LGD) which uses gini to test the quality of slot and also consider number of intervals.

Furthermore, the data which are gathered from sensors are also not always relevant and important to activity recognition learning algorithm because some sensor-recording values are weakly related to activity detection. In real world situation, using all features may produces adverse effect on training process, because of redundant or noisy features. Because of it, feature selection is needed to improve the quality of the data. Furthermore, feature selection is also able to reduce the model size and it is very important as that the resource and space in the mobile device is limited. We see each sensor-value pair, which is the outcome after discrete step, as a feature. In this paper, we use entropy-based feature selection [14] approach to select the feature rather than solely choosing it based on its coverage only. The selection of the feature is also involving learning algorithm such as Naive Bayesian which is used to evaluate selected features. One approach is using one-cut entropy threshold as the selector of the features and cut every features which not satisfy the threshold. We also propose another approach that is greedy-iteration technique which is able to select the features using several iterations. Because of it, greedy approach is able to preserve some feature which is being eliminated in the one cut approach.

Our main contributions in this paper are:

– We propose a framework to recognize human activity behavior by using smart phone sensors. We focus more on life activities instead of motion activities, as it is more meaningful for human.
– We designed a LGD approach to discretize the feature data by using Gini index and number of intervals. This approach has good compatibility with the rule-based classifier and it can improve the accuracy of classifier.
– We designed an iteration-based feature selection approach with the help of entropy as a measure which is able to select several features that have special characteristics, to overcome the limitation of using one-cut approach.

2 A Framework of Activity Inference

In this paper, we use android phone as a platform to detect human activity. Our collected data is a sequence of sensor data d in the form $d = < t, S >$ where t denotes a timestamp, and S denotes a sensor ID and its values. Our output in mobile device is the activity of user. Figure 1 shows the overall architecture of our activity recognition system. As figure shown, this work is consist two parts, off-line part and on-line part.

In the off-line phase, we do the training work. There are four main tasks in this phase called Feature Extraction, Data Discretization, Feature Selection and Model Construction. In Feature Extraction, every sensor has different sampling rate, so each record data d do not necessarily always have complete sensor data. We fill the missing sensor data by adding *Non-Data*. After that, we extract four types of feature called motion feature, location feature, temporal feature and

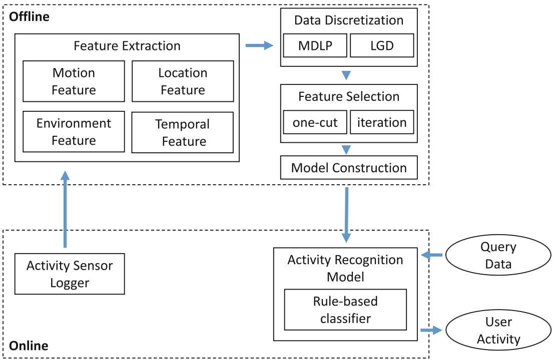


Fig. 1. Proposed framework of activity recognition in smart phones.

environment feature. Each feature contributes different meaning in varied activities and users. After extracting features, we need to discretize those features as our activity recognition algorithm only processes categorical data. In this paper, we adopt one method called MDLP as comparison and propose one method called LGD. MDLP uses the idea of minimum description length to cut the interval and let each interval has significant meaning. LGD consider the data distribution and feature's cooperation. Each method has its own advantages. The detail of these two methods will be introduced in Sect. 4. Another consideration is that some features are meaningless to some user. Thereby we also do feature selection to let the features personalized and even more it is also able to reduce the model size. In feature selection phase, we use one-cut approach and memory-iteration-based approach to select the features. In the last step, we construct rule-based classifier. Rule-based classifier has some advantages which make it suitable for recognizing activity in mobile device. The detail of the model and its advantages will be introduced in Sect. 3.

In the on-line phase, we do the recognizing work. Mobile device uses the result from feature selection phase to determine which feature should be collected and sent to the classifier. The classifier use the model, which is built in the off-line phase, to recognize the user activity and show in the application. For example, Table 1 show how the model is in mobile device. If the sensors collect that the value of GPSspeed is 5.5 m/s and the value of AccelSd is 2.7 m/s^2, the application in mobile device will recognize that the user's current activity is transportation.

Table 1. Example of rule-based classifier model

Priority	Rule
1st	GPSspeed = (0–6] and AccelSd = (0–4] − > Transportation
2nd	GPSx = (0.1–0.4] and GPSy = (0.0–0.2] − > Working
3rd	AccelAvg = (6–8] and AccelSd = (2–4] and GyroSd = (2–4] − > Sporting

3 Activity Recognition

In this section, we introduce what kind of model is suitable for smart phones by analyzing its characteristic. Then we introduce the challenge of using this model and how we conquer it. Finally, we introduce how to build model and set the parameter.

3.1 Recognition Model for Smart Phones

There are lots of ways to build recognition model. Some of them belong to machine learning, like SVM and Neural Networks. Some of them belong to traditional classification, like bayesian probability model, decision tree and k-Nearest Neighbor (k-NN). Each of them has its own advantages and disadvantages. Determining a good classifier in mobile device itself is a challenge.

There are some characteristics that are required to use classification model in the smart phones. First, the model size need to be as small as possible so that the recognition model can be built in low layer architecture such as *sensor hub*, which is a microcontroller unit that help integrate data from different sensors and process them [7]. Since some sensor hub has only 16 KB RAM so model size is an important issue. Another issue is that in order to enhance user experience to its maximum, then the model need to recognize the activity as fast as possible. In this paper, we find that associative classifier satisfies both request above. The recognition model we used is Classification Based on Associations (CBA) [8]. It is composed by ordered associate rules which is shown in Table 1. We use this model by considering several reasons which is explained below.

- Rule-based uses straightforward approach to recognize the activity in on-line step. CBA uses only rule matching to classify so that it can reduce the time and resource for activity recognition.
- In most case, model size is smaller than bayesian probability model, decision tree and SVM, so it is suitable to be put in smart phone.
- Rule-based model is user friendly, so user can see the model to realize their behavior and modify it.

3.2 Model Building and On-line Recognition

Associative classifier is suitable for mobile device but there is still a challenge about how to use sensor data to build CBA model. In this paper, we first discretize the raw data into some intervals. We can see each interval and its corresponding feature as an item. So the builder can use this item and its label to construct the CBA model.

Discretization method also determines the quality of recognition model. Good discretization method can reduce the model size and save model building time. Details about discretization methods that we use are explained in Sect. 4.

In off-line step, we construct the model. We use discretized data to generate frequent and credible associated rules. The parameter of minimum support is configured 0.025 that can let model size small and accuracy high enough. After generating all qualified rules, we build the classifier model by removing all rules which cover no data.

In on-line step, mobile phone has the information of which are useful features (by feature selection) so that mobile device will only open the sensor that its features are useful. This can reduce the power consumption to log useless data. After collecting user's sensor data, like off-line step, we extract the features from raw data. The classifier model try to match the rules from the model and extracted features to recognize user's activity.

4 Features Extraction and Selection

In the following section, we will explain more about collecting the data inside the smart phone, extract the features, discretize the feature to alter continuous values into categorical type and do feature selection to filter out irrelevant features.

4.1 Data Collection and Features Extraction

In our experiments, we create an android apps to log our activities, based on six activity labels, which are: working, entertainment, sporting, shopping, transportation, and dining. The application records all the sensors data from the smart phone and the corresponding activity label. The apps are being used by 50 peoples who are using android 4.2 smart phones in eight months to collect

Table 2. Sensor and its corresponding features

Sensor	Features	Description
Time	Date	Monday, Tuesday,...to, Sunday
	Period	0–24 h
Accelerometer	AccelAvg	Average force of acceleration
	AccelSd	Standard deviation of force of acceleration
Gyroscope	GyroAvg	Average force of Gyroscope
	GyroSd	Standard deviation of force of Gyroscope
Proximity	GyroSd	Average force of Proximity
GPS	GPSx	Average longitude
	GPSy	Average latitude
	GPSspeed	Average speed
Magnetic	MageAvg	Average Magnetic
	MageSd	Standard deviation of Magnetic
pressure	PressureAvg	Average pressure
Light	LightAvg	Average light

the data. Most users log their activity data for at least one month and the most user logs is about four month worth.

Raw sensor data that have been collected which is recorded every single second, can be divided into four types. First is motion sensor (Accelerometer, Gyroscope, and Proximity) which is designed for collecting user's body motion. Second is location sensor (GPS), which collects user location information. Third is environment sensor (Light, Magnetic and Pressure) which logs the environment data around the user. Final one is time sensor which records the time, containing time of the day and day of the week information. For these sensors data, we extract some statistical features such as average and standard deviation. Each sensor data is put together in one row, grouped together in 10 second frame. Several sensor data is also split based on the axis, such as accelerometer, gyroscope, magnetic and GPS. The details of the sensors and their features are shown in Table 2. After extracting statistical features from original raw data, we focus on the distribution given for every feature. Each feature has unique distribution. Each activity label on each feature also show some unique distribution. In the following section, feature distribution is used to select the feature which fit best to infer the activity label.

4.2 Data Discretization

Traditionally, some classification learning algorithms and feature selection assume that attributes are in numerical values. In this paper, our method is also assuming that input for the model is in numerical values. In order to build activity recognition system for mobile device using Rule-based classifier, we need to discretize the data first, as the data which is collected from mobile phone sensors are in continuous values.

Algorithm 1. LGD (Length-Gini Discretization) algorithm

Input: Continuous data to be discretized: S
Output: The discrete data: S'

Define $BestCut = \emptyset$;
Define $NumberOfInterval = 1$;
while $NumberOfInterval < SizeofS$ **do**
    Define $Cutway = S$ equal weight partition into $NumberOfInterval$;
    Define $\alpha = 0$ to 1 (suggest 0.3 to 0.6);
    Define $Gini =$ Gini index after partition data S;
    Define $L_{value} = NumberOfInterval/Numberofthisdata$;
    Define $LGD_{index} = \alpha * Gini + (1 - \alpha) * L_{value}$;
    if $LGD_index > BestLGD$ **then**
        Set $BestCut$ to $Cutway$;
        Set $BestLGD$ to LGD_{index};
    end
    Increase $NumberOfInterval$;
end
return $Cutway$

Our work uses two famous standard (entropy and Gini) to determine where is the best cut point. We do some observation to find how many intervals we need.

By observing the accuracy and model size of CBA under different ways of discretization, the result shows two things. First, cutting few intervals will let each interval blend too much meaning and let this interval useless. Second, although cutting many intervals can let each interval pure, but also it will let that each interval's coverage becomes smaller and lose the opportunity to co-work with other attributes.

Based on observation, we need a standard which consider both number of interval and confusion of each interval. In this paper, we design Interval Length-Gini Discretization (LGD) method which is modified from Gini index. LGD using Gini to consider the confusion of each interval and add a formula to consider number of interval for partitioning. The smaller LGD value is, the better partition is. This algorithm use iterator way to choose the best partition point. Each round raw data will be partitioned into different intervals. Then LGD will calculate the LGD value of this partition. If the LGD value is the smallest one. LGD will record this partition way. After that, number of partition will be increase and go to next round. Algorithm 1 shows the detailed procedure of the LGD. In addition to LGD, we also adopt another discretization method called MDLP [3]. MDLP use the concept of MDL so it determine to cut if and only if information gain is more than the loss to depict the cut point. We implement a recursive algorithm to discretize our sensor data.

Both LGD and MDLP have its own advantages. While MDLP has lower training time, it has lower accuracy performance. Because MDLP uses the concept of MDL, so it accepts data loss. On the other hand, LGD is designed to improve the accuracy of associative classifier so it will have higher accuracy in the cost of having longer training time.

4.3 Feature Selection

In this paper, we use entropy as a measure to calculate the correlation between each feature and class label. As the entropy values gets bigger, feature becomes more impure so the correlation between the feature and the class label become lower. The information about feature impurity shows us that the feature is able to distinguish the class label well or not. For each feature, which are in sensor-value pair, entropy information will be added so we can determine the quality of each feature. We will use one-cut feature selection and iterative-based feature selection, where one-cut only consider one threshold to select the feature, and memory-iteration-based feature selection will use several iterations to select the feature better.

One-cut approach is simply using entropy threshold and remove the feature which has entropy above the threshold. In the different threshold, the result of feature selection may result different output. The ideal situation is where we only select few features but the performance of learning algorithm is remarkably high. For example, selecting entropy 0.5 as the threshold on specific user may remove up to 70 % on the features but is able to only reduce the accuracy by 15 %. In general, one-cut approach has the advantages of having fast and simple implementation. Another approach, memory-iterative-based is using several iteration to select the features by also using locally weighted naive Bayesian [4] as the learning algorithm. The advantage of using Bayesian is that it only requires small amount of training data to estimate the parameters (i.e. mean and variance of the variables) necessary for classification. Because independent variables are assumed, only the variance of the variables for each class needs to be determined. For every iteration the number of features are decreasing as the threshold goes lower or tighter. This process is repeated until entropy threshold reaches minimum entropy value. The process can also stop before entropy threshold reaches minimum level if the overall performance is decreased to some level. In this paper, we use 0 as the minimum entropy value and 80 % of original accuracy as the minimum overall performance requirement. Another issue is that in some particular thresholds, certain selected features may affect the performance of learning algorithm significantly. We also found out that some features can be considered special, as they contradict each other. Feature is having high entropy which is considered as bad feature because they are impure, but they have also many occurrences in the sensor-value pair set. These certain features lead us to use temporary variable inside iteration to obtain better feature subset result. The process can be explained as follow. We remove several features in each entropy threshold. Removed features are first stored in temporary variable first. Selected features will be first evaluated using learning algorithm and then if the performance of learning algorithm is significantly reduced compared to the previous iteration, then the removed features will be added back to feature list and tag them as special feature. The detail memory-iteration-based feature selection is explained in Algorithm 2.

Algorithm 2. One-step-memory iteration-based feature selection

Input: Feature sets: S, starting threshold E, entropy deviation D, reduction tolerance T
Output: Feature subsets: S'

```
Define Temporary = ∅;
Set Acc = 0;
while threshold > 0 do
    foreach f in S do
        if f.entropy > threshold OR f.tag is false then
            Remove feature f from S ;
            Add feature f to Temporary ;
        end
        Evaluate feature subset using learning algorithm ;
        prevAcc = Acc ;
        Acc = current accuracy of learning algorithm ;
        if prevAcc − Acc > tolerance then
            foreach temp in Temporary do
                set temp.tag to be true ;
                put back temp to S ;
            end
        end
        threshold = threshold − deviation ;
        Clear Temporary ;
    end
end
```

Table 3. Statistics information of the real dataset

	Average	max	min
Number of day to collect of a user	44	136	7
Collected data size of a user	634 MB	2289 MB	1 MB

5 Experiments

In this section, we evaluate the performance, including accuracy, efficiency and space usage, of the proposed recognition method and compare it with other baseline classifier methods. We also explain the impact of using different discretization methods. Finally, the influence of feature selection is delivered.

5.1 Experimental Environment

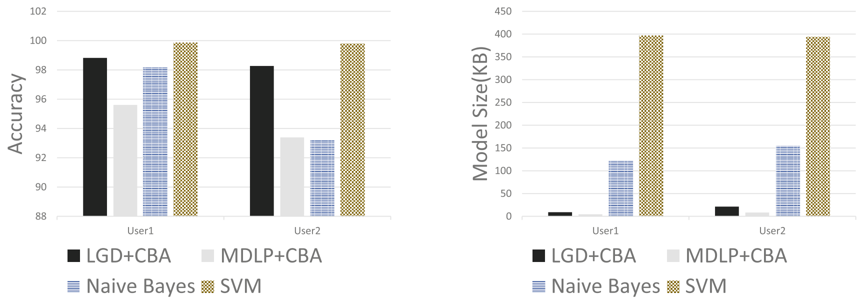
First of all, we introduce the environment of our experiments, including the characteristics of the dataset we used and the measurements to evaluate the recognition performance.

Dataset Description. In this paper, we conduct extensive experiments on real dataset which is collected from volunteers. We implement our logging program on the Android 4.2 platform to record the user's activity and its sensor data. The logger will send the record to server every 10 minutes. For this dataset, we collect 50 participant's activity behavior from May 2013 to December 2013. In our experiments, each users log their activity data for at least one week and the most user log is about four month worth. Each user data is separated as 80 % for training and 20 % for testing. Table 3 shows the statistical information of our dataset.

Compared Methods. Although we introduce some existing activity recognition method, they have different setting with our experiment, such as using different set of sensors and limited equipped position of the mobile device. Therefore, we conduct two famous method, Naive Bayesian and SVM, as our comparison. Naive Bayesian is commonly-used algorithm as it only uses probability model to predict the class label. SVM is another commonly-used algorithm which has high accuracy in most of the experiment cases. We implement these two methods as our comparison. The testing platform and input data is the same as our method. We use WEKA library which provide many famous machine learning algorithm to implement both Naive Bayesian and SVM method. For our method, we also evaluate the performance under different discretizations and feature selections.

5.2 Experimental Results

After we conduct the experiment, we are able to present that our proposed method has an advantages compared to another commonly-used algorithms.



(a) Accuracy Comparison (b) Model Size Comparison

Fig. 2. Different Recognition Method Comparison where CBA's minimum support = 1 % and SVM and Naive Bayesian use default setting

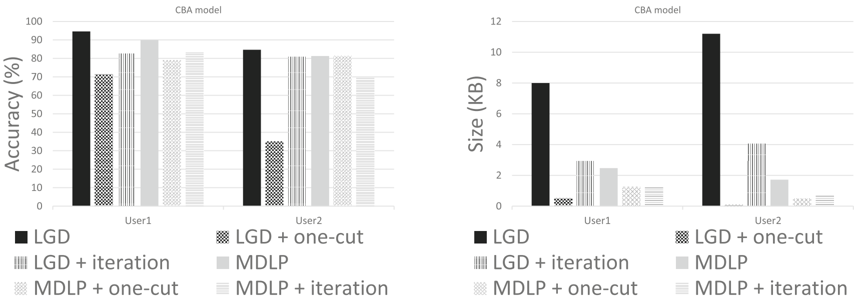
As we concern more to activity recognition in the mobile device, we would desire low model size more, which in some case need to sacrifice some accuracy performance.

Accuracy and Resource Comparison. We first show the experiments result of accuracy on real user dataset. Figure 2(a) show the accuracy performance of our proposed algorithm with two other algorithms. In general, the accuracy of our method are close to SVM and slightly better than Naive Bayesian.

Then, we evaluate the model size on real user dataset. Model size is important issue for mobile. So that the model can be built in sensor hub, model size need small enough. Figure 2(b) show the model size of our proposed algorithm with two other algorithms. For testing the model of SVM and Naive Bayesian, we use WEKA to store the model in the hard drive and remove all default explanation by WEKA. In terms of model size, our proposed method take a runaway lead.

Impact of Features Selection. Figure 3(a) show the influence of accuracy under feature selections and Fig. 3(b) show the influence of model size. As figure above shown, the experiment show that feature selections can reduce half of model size under not reducing too much accuracy. In our observation, if the user has complex lifestyle, memory-iteration-based approach show more stable result, which have smaller performance reduction compared to one-cut approach.

Parameter Studies. Discretization only has one parameter α which balances the gini index and the number of intervals. Table 4 show the model's (i.e. LGD+CBA) accuracy under different α. The value of α should be between 0.3 to 0.6 because (1) it have good accuracy and (2) setting α too low will increase little accuracy but increase much run time. In model building step, there is one parameter called minimum support s. For these parameter, there are much existing research to discussion. In this paper, we suggest that s set to be 1 % to let accuracy higher or set to be 2.5 % to let model size smaller.



(a) Accuracy Comparison (b) Model Size Comparison

Fig. 3. Different Feature Selection Method Comparison where CBA's minimum support = 2.5 %

Table 4. Accuracy Comparison under different α where CBA's minimum support = 2.5 %

α	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
accuracy	50.84	94.76	95.27	94.93	94.59	94.26	94.42	94.42	94.59	93.75	92.91	88.85	89.53	88.01	88.18	83.61	84.63	84.29	84.29	83.61	83.61

6 Conclusion

In this paper we proposed a complete framework ACF for activity recognition using smart phone. In order to deal with continuous value data and to reduce model size while preserving the accuracy performance, we apply two phases of preprocessing: discretization and feature selection. For discretization, we use two approaches, LGD which use Gini index to consider the confusion of each interval and MDLP which use the concept of MDL (Minimum Description Length) to cut if and only if the information gain is more than the loss to depict the cut point. Feature selection technique that we use here including one-cut approach and memory-iteration-based approach. One-cut approach has better running time performance while memory-iteration-based approach has better accuracy performance. In our experimental study, recognition framework that we proposed has also the advantage on smart phone usage as it has smaller model size and considered having high accuracy compared to two other commonly-used algorithms.

Acknowledgement. Wen-Chih Peng was supported in part by the National Science Council, Project No. 102-2221-E-009-171-MY3 and 100-2218-E-009-016-MY3, by hTC and by Academic Sinica Theme project, Project No. AS-102-TPA06.

References

1. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: IE, pp. 214–221 (2012)

2. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: ICCCN, pp. 65–72 (2005)
3. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI, pp. 1022–1029 (1993)
4. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive bayes. CoRR, abs/1212.2487 (2012)
5. Gini, C.W.: Variability and mutability, contribution to the study of statistical distributions and relations (1912)
6. Honda, D., Sakata, N., Nishida, S.: Activity recognition for risk management with installed sensor in smart and cell phone. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part III, HCII 2011. LNCS, vol. 6763, pp. 230–239. Springer, Heidelberg (2011)
7. Lidstone, R.: Website news: Atmel sensor hub mcu helps enhance samsung galaxy s4 user experience, battery life. MobilityTechzone, May 2013
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining, pp. 80–86 (1998)
9. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Ann. Math. Artif. Intell. **41**(1), 77–93 (2004)
10. Rashidi, P., Cook, D.J.: Mining sensor streams for discovering human activity patterns over time. In: ICDM, pp. 431–440 (2010)
11. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV, pp. 1036–1043 (2011)
12. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. ICPR **3**, 32–36 (2004)
13. Stikic, M., Larlus, D., Ebert, S., Schiele, B.: Weakly supervised recognition of daily life activities with wearable sensors. TPAMI **33**(12), 2521–2537 (2011)
14. Zhu, S., Wang, D., Yu, K., Li, T., Gong, Y.: Feature selection for gene expression using model-based entropy. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(1), 25–36 (2010)