Human Activity Recognition with Smartphones

Rao Fu
Computing Science
Simon Fraser University
raof@sfu.ca

Yao Song
Computing Science
Simon Fraser University
songyaos@sfu.ca

Weipu Zhao
Computing Science
Simon Fraser University
weipuz@sfu.ca

Abstract

This report focuses on improving classification accuracy and reducing computational complexity for human activity recognition problem on public datasets UCI and WISDM. We discussed the benefits of getting access to smartphones in the filed of HAR research. Our experiment indicates that combining AdaBoost M1 algorithm with C4.5 contributes to discriminating several common human activities. Moreover, we showed that it is feasible to reduce computational complexity and achieve high accuracy at the same time by applying correlation-based feature selection.

1 Introduction

Human activity recognition (HAR) is nothing new. Image based HAR has been studied in the field of computer vision for a long time. The goal of HAR is the same as that of a broader concept called context-aware computing or ubiquitous computing. In ubiquitous computing, the sensor collects data from the user and tries to assist the user with the task. HAR has a wide application in medical area, military field, family entertainment and personal daily life.

In [1], automatic activity recognition techniques are used in the soldier assist system to help soldiers with their after action reports. In [2], HAR techniques are used to help hospital staff with their daily working by estimating their working activities. In [3], daily activities are learned in order to detect abnormality of the person so that caregiver will be alerted if necessary. There are tons of research talking about Microsoft Kinect, which works by capturing the human movement and gestures. Some running shoes have integrated motion sensors to provide performance feedback for athletes [4]. It is obvious that HAR is getting more and more importance in every aspect of our life.

Thanks to the efforts of researchers, basic activities such as sitting, walking, running can be recognized with high accuracy given multiple sensors on the subject. However, it is not practical to have multiple sensors on the user in daily life. Recently, activity recognition using wearable consumer electronics has attracted more interests. The most popular one is smartphone which boosted in recent years. These days new smartphones are all equipped with multiple sensors for different purposes. People are carrying these "sensors" nearly all the time. These smartphone sensors, especially the accelerometer give us several obvious advantages in the HAR research.

First, smartphones allow the data to be collected anywhere and anytime in the day from the subjects with less obtrusiveness. The subject just needs to carry the smartphone while performing the required activities. In this way, we can obtain realistic data from the real world daily activities. It is much better than the lab controlled data collection method, which usually requires many sensors on different parts of human body. Lab controlled data tends to generate fake high classification accuracy compared with realistic data. As shown in [5], the accuracy dropped nearly 30% when changing from lab data to real data. Second, we get access to more data by using smartphones as data source. In previous researches, due to lab equipment and other constraints, people usually study on less than 10 subjects, which significantly limited the generalization of the research results. With smartphones as sensors, it is easy to have 30 subjects with different ages. People are using their phones all the time and the data can be accumulated over time. Overall, we can train the model based on a larger and more realistic dataset with smartphone sensors. However, the difficult problem is we only have one sensor source now and the position of the sensor is not accurately controlled by the researchers.

It has been shown by previous researches [6] that when the number of sensors decreases, the recognition accuracy tends to decrease as well. This fact is proved in smartphone based HAR research. In [7], the detecting accuracy for some activities, such as walking upstairs or walking downstairs is quite low. On the contrary, with multiple sensors, the recognition of such basic activities is not a difficult job [8]. It is also noted that the position of the sensor on the body will affect the recognition task. All this motivates us to study the smartphone based activity recognition problem.

From our point of view, there are mainly two ways to deal with the single sensor problem arising in the smartphone based HAR. The first method relies on the data collection and feature generation and selection step. If we can get access to more data, the model obtained after the training process is more likely to give a better prediction on the future testing data. The second method does not focus on the dataset, but the classification algorithm which trains the dataset. Researchers have tried different learning algorithms on their own datasets.

However, currently, different research groups use data collected by themselves from various sources. Moreover, most data are not available to the public. Among those available, raw data is seldomly provided, only transformed feature data is accessible. This makes it impossible to aggregate the data from different groups. It is also impossible to compare the results presented by different research groups. This lack of data situation stops us from generating new features based on raw data. However, given publicly available dataset, we can still try to target the important features to gain some insights into the smartphone based activity recognition problem.

In this paper, we focus on the second approach. We make use of publicly available datasets online and try to improve the classification accuracy by choosing the proper learning algorithm. We also reduce the computational complexity by employing feature selection algorithm. The rest of the paper is organized as follows: Sectoin 2 talks about the main approach we have utilized in the learning process. Section 3 presents and discusses our simulation results. Finally, we conclude our work in section 4.

2 Approach

2.1 HAR Procedures

According to [4], HAR process can be divided into several stages. We first obtain raw data from sensors. In the smartphone case, the data may come from the accelerometer. The raw data need to be preprocessed for classification training algorithm . For smartphone accelerometer, time series data were segmented into fractions. Features are then generated and selected based on raw data. Given enough data samples, we can train our model using proper learning algorithms.

2.2 Datasets

In this report, we use two datasets publicly available online. The first dataset is from the WISDM group [7]. The data was collected from users carrying an Android smartphone with a data collection app in the lab environment. The Android phone is put in the subject's front leg pocket and five activities including walking , jogging, walking upstairs, walking downstairs, sitting, and standing, were monitored. The data was sampled at the rate of 20Hz. Then the raw data is segmented into 10 second segments without overlapping. So there are 200 data points in each segments. From 6 basic features, 43 features in total are generated from the raw data. Details about the features can be found in the original paper.

The second dataset is from [9]. The data was collected from 30 volunteers with the age range of 19-48 years old. Each subject performed six activities wearing a smartphone on the waist. The six activities are walking, walking upstairs, walking downstairs, sitting, standing and laying. The data was sampled at maximum rate of 50Hz. The segmentation method is different from the first dataset. The window size is 2.56 seconds. So there are 128 data points in each segment. However, the segments are 50% overlapped.

2.3 AdaBoost M1

Boosting method was first introduced by Yoav Freund and Robert Schapire [10]. As described in [11], it works by combining several weak learning algorithms with weights into a strong learning algorithm which yields higher accuracy than each base learning algorithm. AdaBoost, short for "adaptive boost", is an improved version of the original boosting algorithm in the way that it will adapt to the individual learner's error rates. In the WEKA [8], Adaboost M1[12] method is implemented.

Yoav Freund and Robert E. Schapire proves that when extending AdaBoost M1 to a multiclass classification problem, each weak learner has prediction accuracy more than 50%. In our experiment shown in section 3, the overall accuracy achieves over 80% when we only apply C4.5 algorithm. Therefore, using C4.5 as our weak hypothesis in AdaBoost M1, it is plausible to improve accuracy. The whole algorithm [12] is as below.

Algorithm 1 AdaBoost M1

Precondition: ***input*** $(x_1, y_1), ..., (x_m, y_m)$, $y_i \in Y = \{1, ..., k\}$

```
function AdaBoost.M1($x, y$)
    $D_1(i) \leftarrow 1/m$
    for $t \leftarrow 1$ to $T$ do
        call weak learning algorithm with $D_t$
        compute hypothesis $h_t(x)$
        calculate $\epsilon = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$
        if $\epsilon > 1/2$ then
            $T \leftarrow t - 1$
            abort the loop
        else
            $\beta_t \leftarrow \epsilon/(1-\epsilon)$
        if $h_t(x_i) = y_i$ then
            $D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} \times \beta_t$
        else
            $D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} \times 1$
    return $h_{fin}(x) = argmax_{y \in Y} \sum_{t:h_t(x)=y} \log \frac{1}{\beta_t}$
```

The algorithm assigns weight distribution D_t to training instances. Initially, D_1 is equally distributed among samples. It runs T iterations. For each iteration, it focuses on hypothesis misclassifications by applying the user-specified weak learning algorithm with distribution D_t and for each misclassification, accumulating its distribution $D_t(i)$. The accumulation is used to obtain weights for correct classifications. Then we compute weighted distributions among instances and renormalize the results for next iteration distributions. In Algorithm 1, it is obvious that correct classification will always get a weight no more than 1, so its weight is smaller than incorrect classification. It explains that AdaBoost M1 puts the most weight on the "hardest" sample given a weak learning algorithm. Finally, for a given sample x_i, we consider all rounds of hypotheses and their predictions for labels. We choose the label $y_i \in Y$ that maximizes the sum of weights where the output of hypothesis $h_t(x_i)$ is equal to y_i and set final hypothesis $h_{fin}(x_i)$ output y_i.

2.4 Correlation-based Feature Selection

In practice, a high dimension of features might result in "curse of dimensionality". Moreover, some of these features might be redundant attributes and will mislead the modeling algorithm. Furthermore, taking irrelevant features into consideration might lead to over-fitting on test data because it decreases the importance of good quality features. In either case, a high dimension of features increases complexity and has a negative effect on final result. Hence, feature selection becomes essential in our experiment since we have 561 features in the original dataset.

Table 1: Result for UCI HAR dataset: (number of attributes: 561, training instances: 7352)

Method	**Test**	**F-measure accuracy (%)**						
	option	**Walking**	**Upstairs**	**Downstairs**	**Standing**	**Sitting**	**Laying**	**Overall**
MC-svm	testset	91.2	77.5	77.5	95.0	96.0	100.0	89.3
MC-HF-svm	testset	91.1	76.6	76.7	94.9	95.5	100.0	89.0
C4.5	10cv	95.2	94.5	94.0	93.8	93.3	100.0	95.3
C4.5	testset	80.7	72.2	79.1	83.0	79.6	100.0	82.9
AdaBoost(10it)	10cv	99.2	99.2	99.0	96.1	95.8	100.0	98.1
AdaBoost(10it)	testset	93.1	88.6	92.1	87.6	85.0	100.0	99.2
AdaBoost(100it)	10cv	99.6	99.7	99.6	98.4	98.1	100.0	99.2
AdaBoost(100it)	testset	96.2	91.8	94.0	90.5	89.1	100.0	93.6

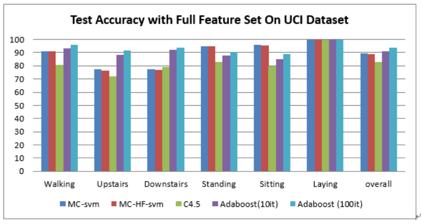


Figure 1: Test Accuracy with Full Feature Set on UCI Dataset

Correlation-based Feature Selection (CFS) for machine learning is introduced by Mark A. Hall [13]. CFS is a filter algorithm that ranks subsets of features by a correlation based heuristic evaluation function as shown below [13]:

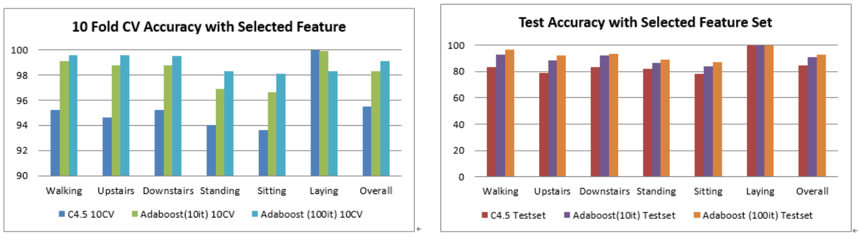
$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}}, \qquad f \in S \tag{1}$$

M_s is the measure of subset S containing k values, while $\bar{r}_{cf}$ is the mean correlation for feature-class, and $\bar{r}_{ff}$ is the feature-feature mean correlation. CFS chooses the subset of features which has the highest measure. The chosen subset holds the property that features inside this subset have high correlation with the class and unrelated with each other. CFS assumes that features are conditionally independent on the class. However, it can still work well when there exists slight feature dependence.

3 Experiments

3.1 Comparison of Classification Algorithm

As described in section 2.3, Adaboost M1 is a power algorithm to improve accuracy of classification. We designed several experiments to compare the result of Adaboost algorithm on the two selected datasets with their original methods. In particular, we choose C4.5 decision tree as the weak learner of Adaboost M1. Table 1 and Figure 1 shows the detail accuracy of certain classification method on dataset 2, UCI dataset[9]. Multiclass svm and Multiclass Hardware Friendly SVM was used in [9] on dataset 2 with 70% training set and 30% test set. We compute the accuracy of simple C4.5 algorithm and the Adaboost M1 using C4.5 as weak learner. Additionally, [14] stated that 100 is a proper number of iteration for Adaboost M1 thus we choose 10 and 100 as two different iteration numbers of Adaboost in our experiment. In order to be generally comparable, for each method we use both 10-fold cross validation and test set as test options. Similarly, same approaches and 10-fold cross validation was apllied to dataset 1, WISDM dataset[7]. Table 2 shows the results comparing to Multilayer Perceptron which is originally used in [7].



(a) 10 fold CV Accuracy with Selected Feature

(b) Test Accuracy with Selected Feature Set

Figure 2: 10 fold CV Accuracy and Test Accuracy with Selected Feature Set on UCI Dataset

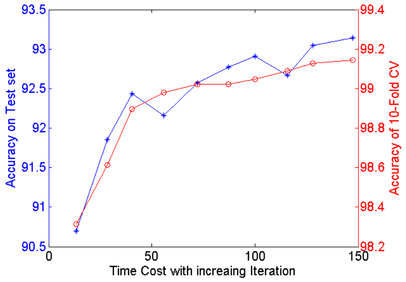


Figure 3: Test and CV Accuracy Curve with Increasing Computation Time

3.2 Result Comparison Before and After Feature Seletion

It is very clear that in both datasets, Adaboost M1 outperformed other methods, especially with 100 iterations. But in dataset 2, with 561 attributes as input feature vector, the computation time for building model is relatively high, which is impossible to be used in real world application. Thus CFS feature selection method is employed to reduce feature dimension. After feature selection, same classification methods were tested with respect to time consumption and classification accuracy. Table 3 shows the comparison of classification time consumption of full feature dataset and selected feature dataset. It shows that running an additional time consuming feature selection algorithm is worthwhile for a complex method such as AdaBoost. The total time spent for feature selection and model building is nearly 10 times less than building model on full feature set. What's more important is the accuracy of selected feature set is comparable to that of full feature set. Table 4 shows the accuracy comparison of two feature sets. The accuracy of AdaBoost with 100 iterations on selected feature set only decreased less than 1%. Detailed accuracy results on selected feature set are listed in Figure 2 and Table 5.

3.3 Proper Iterations on AdaBoost M1

The feature selection nethod significantly reduced the computation complexity of AdaBoost algorithm. Since the iteration number of AdaBoost is very important and highly related to classification accuracy, choosing a proper iteration number with reasonable computation time is critical. Table 6 shows the comparison of test accuracy and 10 fold cross validation accuracy with different iteration numbers from 10 to 100. Figure 3 shows the relationship curve between accuracy and time cost. With increasing number of iteration, AdaBoost M1 does not have over fitting problem. Both test accuracy and cross validation accuracy increased with higher computation time and the highest accuracy occurs at the largest iteration number. Time consumption and accuracy tradeoff might be needed especially in mobile computing.

Table 2: Result for WISDM dataset:(number of attributes: 43, training instances: 5418)

Method	Test option	F-measure accuracy (%) Walking	Jogging	Upstairs	Downstairs	Sitting	Standing	Overall	Model build time(s)
Multilayer perceptron	10cv	95.9	99	71.2	68.0	98.4	93.4	91.2	213.99
C4.5	10cv	94.6	95.8	68.0	65.7	97.5	97.2	89.3	1.27
AdaBoost	10cv	97.8	98.6	80.9	77.8	98.2	95.9	94.0	8.02
AdaBoost(100it)	10cv	98.7	98.8	83.7	80.6	98.5	96.7	95.1	72.08

Table 3: Time Consumption with Different Feature Sets

Method	Test option	Model build times(s) with 561 features	Model build time(s) with selected 50 features	Total time(s) of feature selection and model build
C4.5	10cv	14.65	1.21	55.02
C4.5	testset	14.65	1.21	35.60
AdaBoost	10cv	172.78	13.34	74.25
AdaBoost	testset	172.78	13.34	55.26
AdaBoost(100it)	10cv	2080.41	147.38	255.68
AdaBoost(100it)	testset	2080.41	147.38	215.32

4 Conclusion

This project presents experiments of Adaboost.M1 for smartphone-based human activity recognition. Detailed accuracy of Adaboost.M1 and several other methods was compared on two different datasets. In both datasets Adaboost.M1 outperformed commonly used classifiers. To further reduce the feature dimension and make the algorithm applicable, CFS based feature selection method was employed. The selected feature set have remarkable time efficiency along with comparable classification accuracy. Experiment was also performed to show that training with Adaboost.M1 was robust to overfitting.

Contributions

Rao explored AdaBoost M1 and CFS applied in WEKA. Yao did literature review and wrote the introduction. Weipu designed and performed experiments, gathered and processed experiment results.

References

[1] Minnen David & Tracy Westeyn & Daniel Ashbrook & Peter Presti & Thad Starner. (March-April2007) Recognizing soldier activities in the field. *In 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pp. 236-241. Springer Berlin Heidelberg.

[2] Sanchez D. & Tentori M. & Favela J.. (March-April 2008) Activity Recognition for the Smart Hospital. *Intelligent Systems, IEEE*, vol.23, no.2, pp. 50,57.

[3] Duong T.V. & Bui H.H.& Phung D.Q.& Venkatesh S.. (June 2005) Activity recognition and abnormality detection with the switching hidden semi-Markov model. *Computer Vision and Pattern Recognition 2005, CVPR 2005, IEEE Computer Society Conference*, vol.1, pp. 838,845, vol.1, pp. 20-25

[4] Bulling Andreas & Ulf Blanke & Bernt Schiele. (2014) A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR) 46*, no.3 (2014): 33.

[5] Foerster F. & Smeja M. & Fahrenberg J.. (September 1, 1999) Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, Volume 15, Issue 5, pp. 571-583

[6] Reiss A. & Stricker D. (August 30 - September 3, 2001) Introducing a modular activity monitoring system. *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pp. 5621,5624.

[7] Jennifer R. Kwapisz & Gary M. Weiss & Samuel A. Moore. (2010). Activity recognition using cell phone accelerometers. *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10)*, Washington D.C..

[8] Mark Hall & Eibe Frank & Geoffrey Holmes & Bernhard Pfahringer & Peter Reutemann & Ian H. Witten. (2009) The WEKA data mining software: an update. *SIGKDD Explorations*, vol.11, issue 1.

[9] Davide Anguita & Alessandro Ghio & Luca Oneto & Xavier Parra & Jorge L. & Reyes-Ortiz. (December 2012) Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain.

[10] Freund Yoav & Robert E. Schapire. (1995) A desicion-theoretic generalization of on-line learning and an application to boosting. *In computational learning theory*, pp.23-37. Springer Berlin Heidelberg.

[11] Freund Yoav & Robert E. Schapire & N. Abe. (1999) A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14, no.771-780 (1999): 1612.

[12] Freund Yoav & Robert E. Schapire. (1996) Experiments with a new boosting algorithm. *In ICML*, vol.96, pp. 148-156. [13] Hall M. A. (1999) Correlation-based feature selection for machine learning. Doctoral dissertation. The University of Waikato. [14] Reiss A.& Hendeby G. & Stricker D. (2013). A competitive approach for human activity recognition on smartphones. *In ESANN 2013*, pp. 455-460

Table 4: Result on Reduced Feature sets: (number of attributes reduceced to 50, training instances: 7352)

Method	**Test**	**F-measure accuracy (%)**						
	option	**Walking**	**Upstairs**	**Downstairs**	**Standing**	**Sitting**	**Laying**	**Overall**
C4.5	10cv	95.2	94.6	95.2	94.0	93.6	100.0	95.5
C4.5	testset	83.4	79.1	83.6	81.8	78.5	100	84.6
AdaBoost(10it)	10cv	99.1	98.8	98.8	96.9	96.6	99.9	98.3
AdaBoost(10it)	testset	92.7	88.5	92.0	86.7	83.7	100.0	90.7
AdaBoost(100it)	10cv	99.6	99.6	99.5	98.3	98.1	98.3	99.1
AdaBoost(100it)	testset	96.5	92.3	93.3	89.2	87.1	100.0	93.1

Table 5: Comparison of Accuracy with Different Feature Sets

Overall accuracy	**10-fold with full features**	**10-fold with selected features**	**Test with full features**	**Test with selected features**
C4.5	95.3	95.5	82.9	84.6
AdaBoost(10it)	98.1	98.3	91.1	90.7
AdaBoost(100it)	99.2	99.1	93.6	93.1

Table 6: Accuracy and Time Cost with Increasing Iteration number in AdaBoost

# of iteration	10	20	20	40	50	60	70	80	90	100
Overall 10CV Acc.	98.31	98.61	98.90	99.98	99.02	99.02	99.05	99.09	99.13	99.14
Overall Test Acc.	90.70	91.86	92.43	92.16	92.57	92.77	92.90	92.67	93.04	93.15
Time(s)	13.34	28.28	40.46	55.79	72.20	87.04	100.00	115.60	128.03	147.38