University of California
Los Angeles

# Using Active Learning for Activity Recognition on Smartwatches

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

**Farhad Shahmohammadi**

2016

Abstract of the Thesis

# Using Active Learning for Activity Recognition on Smartwatches

by

**Farhad Shahmohammadi**

Master of Science in Computer Science

University of California, Los Angeles, 2016

Professor Majid Sarrafzadeh, Chair

Advancements in wearable technologies have equipped smartwatches with various sensors. Motion and direction sensors, namely accelerometers, gyroscopes, and magnetometers, have broad applications in human activity recognition due to their ability to detect movements of the hand, wrist, and arm. While many activity recognition algorithms have been proposed in recent years, most studies emphasize the use of inertial sensors in smartphone devices or other body-worn sensors. By comparison, very few works have evaluated the application of smartwatches for activity monitoring applications. In this study, we present a system to detect five daily activities using smartwatches. Our system identifies activities with 91.9% accuracy based on over 540 minutes of data collected from twelve subjects. We also demonstrate that the opportunity to deploy active learning for activity recognition can be a significant advantage of smartwatches over other devices. By collecting personal data from subjects online, active learning improves the accuracy of classifier predictions, while lowering variance between different users.

The thesis of Farhad Shahmohammadi is approved.

Guy Van den Broeck

Douglas S Parker

Majid Sarrafzadeh, Committee Chair

University of California, Los Angeles

2016

# Table of Contents


**1 Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **1**

**2 Data Collection** . . . . . . . . . . . . . . . . . . . . . . . . . . . **4**

2.1 Linear Acceleration Sensor . . . . . . . . . . . . . . . . . . . . . 5

2.2 Rotation Vector Sensor . . . . . . . . . . . . . . . . . . . . . . . 6

**3 Feature Extraction** . . . . . . . . . . . . . . . . . . . . . . . . . . **7**

**4 Results of Supervised Machine Learning Techniques** . . . . . . . . . . **9**

**5 Enhancing the Classifier Using Active Learning** . . . . . . . . . . . . **14**

5.1 Uncertainty Sampling Strategy . . . . . . . . . . . . . . . . . . . 15

5.2 Query by Committee Strategy . . . . . . . . . . . . . . . . . . . 19

**6 Related Work** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **22**

**7 Conclusion** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **27**

**References** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **28**

# List of Figures


2.1 Smartwatch Relative Coordinate System . . . . . . . . . . . . . . . . . . 5

4.1 Accuracy of Different Classifiers on Different Sets of Sensors . . . . . . 10

4.2 F1-Score of Activities Obtained by Extra Trees Classifier on Different Sensor Sets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

5.1 Prediction Certainty for Different Activities . . . . . . . . . . . . . . . . 16

5.2 Active Learner's Work Flow . . . . . . . . . . . . . . . . . . . . . . . . 17

5.3 Simulation of Active Learning with Uncertainty Sampling Using Different Thresholds . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

5.4 Comparison of the Performance of Different Querying Strategies . . . . 20

# List of Tables


3.1 List of Extracted Features . . . . . . . . . . . . . . . . . . . . . . . . . 8

4.1 F1-Score Results of Different Classifiers . . . . . . . . . . . . . . . . . . 12

4.2 Confusion Matrix for Extra Trees Classifier . . . . . . . . . . . . . . . . . 12

4.3 Results Obtained by Extra Trees Classifier for Each Subject . . . . . . 13

5.1 Results Obtained by Active Learner with Threshold Set to 90% . . . . 19

6.1 Review of Studies on Activity Recognition Using Supervised Machine Learning Techniques Based on a Wrist Worn Accelerometer . . . . . . . 25

## Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Majid Sarrafzadeh for all his support during my studies.

# CHAPTER 1

# Introduction

Human activity monitoring has been widely utilized in recent medical and health care research [3, 4, 16]. With ever increasing age of the current population, development of remote health monitoring systems for the elderly seems vital [14]. According to gerontologists, detecting changes in the everyday behavior of the elderly is a valuable criterion for early detection of health problems, and it is often more beneficial than biometric information [33]. Monitoring the elderly's daily activities can play a tremendous role in detecting these changes. In addition, remote monitoring of patients' activities is a valuable resource to keep track of their rehabilitation treatment process [26].

Nowadays most of the cellphones and smartwatches are equipped with motion and direction sensors, which can be used to identify the human activities. Most of the current efforts on this practice is focused on cellphones and there has been less focus on the smartwatches as the primary device [23]. The shortfall of attention paid to smartwatches for activity recognition, in comparison to other devices, could be due to their lack of popularity among the users, which has seen an increase in the past few years, and their limited computation power and battery life and the lack of WiFi support prior to the 5.1.1 Android Wear update. Before this update the smartwatches could only be connected to the Internet via a smartphone. Thus, in the absence of another device with WiFi capability, they could not even play the role of a gateway to offload the computation to a web server. As a result, it was impractical to use smartwatches as the sole device in large scale, for example at a rehabilitation center. In this study we show that smartwatches can identify daily activities quite accurately and they are a great tool for activity recognition. As an activity recognition tool, smartwatches have some advantages over other devices, including smartphones. They minimize the effort of the subjects to keep track of the device and it is also easier for a nurse to keep track of the smartwatches that are worn by patients. Another important advantage of using smartwatches is that they allow us to benefit from active machine learning in order to improve our activity recognition algorithms. Active learning requires querying the user to annotate some unlabeled data points, and hence increasing the training data by the most informative samples [34]. Smartwatches provide an easy way to interact with the user and as a result they can satisfy this requirement perfectly.

In this study we investigate the feasibility of activity recognition using smartwatches. Twelve subjects have participated in our study and we focused on five daily activities: running, walking, standing, sitting, and lying down. Monitoring these activities has been shown to be effective in keeping track of rehabilitation process of patients, especially for chronic diseases. [6, 20]. Our data was recorded without the author's supervision. Considering the broadness of the terms we used, each of which able to refer to a variety of human activities, we asked our subjects to provide us a brief description of the performed action for each category. For the running activity, our subjects either ran on a treadmill or jogged. For the standing activity the subjects were standing on foot and some of them did take a few steps once in a while, but they were mostly stationary. They also reported eating, drinking and speaking during this period. For the sitting activity, our subjects reported typing, playing piano, speaking and eating. Before the experiments we asked our subjects to be active during the sitting period, since resting the hand completely (especially on an arm chair) would result to the exact same hand performance as the lying down activity and, as a result, these activities would have become indistinguishable. For the lying down activity, our subjects laid on their back or stomach, but they were not necessarily still. A number of the subjects reported using their cellphones in this position. For the walking activity, our subjects reported walking with their usual speed on the ground or on a treadmill.

As part of this study we compare the performance of the supervised machine learning techniques on different sensor sets. We introduce two software based sensors implemented by the Android operating system, namely linear acceleration sensor and rotation vector sensor, which are not popular in the literature related to activity recognition. We show that using the aforementioned sensors results in an increase in the accuracy of prediction compared to physical accelerometers. We also study two different methods for applying active learning to our prediction models and demonstrate how these methods can generate personalized models for different subjects, which increases the prediction accuracy, especially for the subjects that have a low prediction accuracy.

The remainder of the thesis is organized as follows. Chapter 2 describes our data collection process and properties of the recorded data. Chapter 3 describes the feature extraction process. Performance of multiple supervised machine learning techniques on data collected from different sensors is discussed in Chapter 4. Chapter 5 describes how active learning can improve the prediction model obtained in Chapter 4 for subject specific use. Chapter 6 describes the related work and Chapter 7 summarizes our work.

# CHAPTER 2

# Data Collection

We developed an Android application to record the data for our task. In this application each user selects his or her username, one of the five activities and a duration of five or ten minutes to perform the activity. Afterwards, the smartwatch starts to record from the sensors with 10 Hz frequency for the specified duration and uploads the recorded data, username, label of the activity being performed and the time of the experiment to a web server. All of our data is recorded using Samsung Gear Live smartwatches.

Twelve subjects participated in our study, including 8 males and 4 females. Their age ranged between 22 to 28 years. Among them, three were left handed and wore the watch on their right hand and the rest wore it on their left hand. Each user was asked to perform walking, standing, sitting and lying down for 10 minutes and to perform running for 5 minutes. We chose a shorter duration for running, since it is inherently easier to distinguish this activity from the other ones, and running for 10 minutes could be physically daunting for our subjects, which in turn would introduce unwanted noise in our data. Since we asked our subjects to perform the activities for a fixed amount of time, our dataset is not skewed towards any subject or activity, and hence we avoid over-fitting our classifier. All subjects performed the experiments without the author's supervision. By asking our subjects to perform the activities for long time intervals, we aimed to remove the unconscious impacts of performing a test on the subjects' behavior, in order for the data to be more realistic. We also removed the first 10 seconds of each interval of 5 minutes, to remove the impacts of starting the experiment from the data.

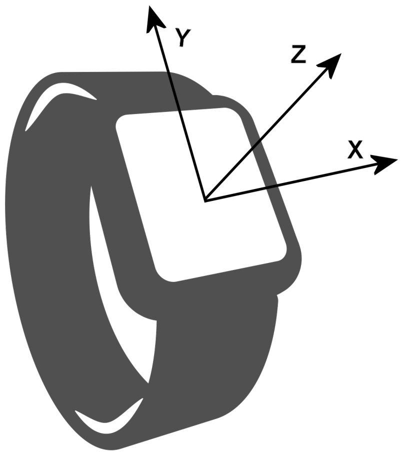


Figure 2.1: Smartwatch Relative Coordinate System

Our Android application recorded the data from accelerometer, linear acceleration and rotation vector sensors. Linear acceleration sensor and rotation vector sensor are software-based motion sensors defined in the Android Open Source Project (AOSP) [1]. In the following sections we briefly discuss these sensors.

## 2.1 Linear Acceleration Sensor

A tri-axial accelerometer is a device which measures the acceleration applied to it in each of the X-axis, Y-axis and Z-axis. Smartwatches are usually equipped with a tri-axial accelerometer which measures the acceleration relative to the device's coordinate system, which can be seen in Figure 2.1. The acceleration applied to the device includes the force of gravity which is useful for determining the orientation of the device, but makes it difficult to find the acceleration created by user's activity. Android's linear acceleration sensor excludes the force of gravity from the accelerometer's measurement and reports the acceleration applied to each axis by the user wearing the watch.

## 2.2 Rotation Vector Sensor

Android's rotation vector sensor is a fused sensor that measures the orientation of the device relative to the East-North-Up coordinates system. It mainly uses integration over the gyroscope to obtain the rotation of the device. In addition, this sensor uses accelerometer and magnetometer sensors to obtain more accurate results. The rotation vector sensor returns device's orientation as a unit quaternion. Quaternions system is a four dimensional extension of the complex numbers system which is shown to be useful for handling spacial rotations [10]. The rotation axis and the rotation angle can be calculated from a unit quaternion. A rotation around the obtained axis by the calculated angle will transform the East-North-Up coordination system to the device's relative coordination system. A unit quaternion is of form $(cos(\theta/2), x * sin(\theta/2), y * sin(\theta/2), z * sin(\theta/2))$ where $\theta$ is the angle of rotation and $(x, y, z)$ is the axis of rotation in three dimensions.

# CHAPTER 3

# Feature Extraction

A common way to extract features from time series data to train supervised machine learning classification algorithms is to divide the time series into fixed length windows [11, 35]. In this study we use windows of length 10 seconds. This window length is shown to be reasonable for activity recognition [15, 36]. Since we collected the data with 10Hz frequency, each window will contain approximately 100 sensor readings. For each window we generate some summary features based on these readings.

An important measure to determine the activity being performed is the magnitude of the acceleration vector which can be calculated from the accelerometer readings. For each window we generate a new time series containing the magnitudes. Afterwards, we generate summary features for acceleration, based on the three time series that were generated by the accelerometer and the magnitude time series.

In order to use the data collected from the rotation vector sensor, we need to transform quaternions to some expressive information. To achieve this goal, we rotate the $(0, 0, 1)$ vector (the unit vector towards sky) by each quaternion in a window to obtain the $(x, y, z)$ coordinates of the unit normal vector of the watch surface. We generated our summary features based on these coordinates. Although unit normal vector of the watch surface does not uniquely specify the orientation of the watch, it has sufficient information for our task. We figured out that including the coordinates of other axes of the relative coordinate system of the watch will make our features noisier, and hence, reduces the quality of classification.

Table 3.1 shows the list of the extracted features. Among our features Dynamic Time Warping distance needs more explanation. Dynamic Time Warping (DTW)

| Features | Sensors |
|---|---|
| Mean | Acceleration & Orientation |
| Standard Deviation | Acceleration & Orientation |
| Skewness | Acceleration & Orientation |
| Kurtosis | Acceleration & Orientation |
| Dynamic Time Warping Distance | Acceleration & Orientation |
| Energy | Acceleration |
| Inter Quartile Range | Acceleration |
| Average of absolute differences between successive points | Acceleration |
| Standard deviation of absolute differences between successive points | Acceleration |

Table 3.1: List of Extracted Features

a well-known algorithm for comparing two time dependent sequences. It finds the optimal alignment between nonlinear warpings of the sequences [24]. We use DTW distance as a feature to recognize repetitive human activities like walking. In order to do so, we divide a 10 seconds interval into two 5 seconds intervals and then compute the DTW distance between these intervals. For repetitive activities like walking this distance is smaller, compared to non-repetitive activities like standing. For acceleration data, we computed this distance using normalized signals to ensure that it is only related to pattern of the signal, and not its magnitude.

# CHAPTER 4

# Results of Supervised Machine Learning Techniques

We described our data collection and feature extraction phases in the previous chapters. In this chapter we compare the performance of different supervised machine learning classification techniques on different sets of sensors, to obtain a predictive model for subjects' activities. In the next chapter we will describe how using active learning can enhance this model. We used Python's Scikit-Learn package [27] as our machine learning tool.

We explored the performance of 5 different classification techniques: Random Forest, Extra Trees, Naive Bayes, Logistic Regression and Support Vector Machines, on 4 different sets of sensors: accelerometer, linear acceleration, rotation vector, and combination of rotation vector with either linear acceleration or rotation vector sensor. Accelerometer sensor has been vastly utilized in the literature related to activity recognition. To evaluate the performance of our models we use leave-one-subject-out (LOSO) cross-validation. That is, for each subject, we train the classifier on the data collected from the other subjects and then we test the classifier on the subject's data. At the end, we report the average performance of classification among all of the subjects. By using this method we ensure that the training and test data are subject-independent. In [5] it is shown that having access to subject specific data can increase the classification accuracy of some activities. Thus, in order to obtain generalizable results we need to preserve subject-independence among training and test data. Supporting this claim, in a recent study [32], it is shown that using 10-fold cross-validation instead of LOSO cross-validation for activity recognition can result in obtaining de-

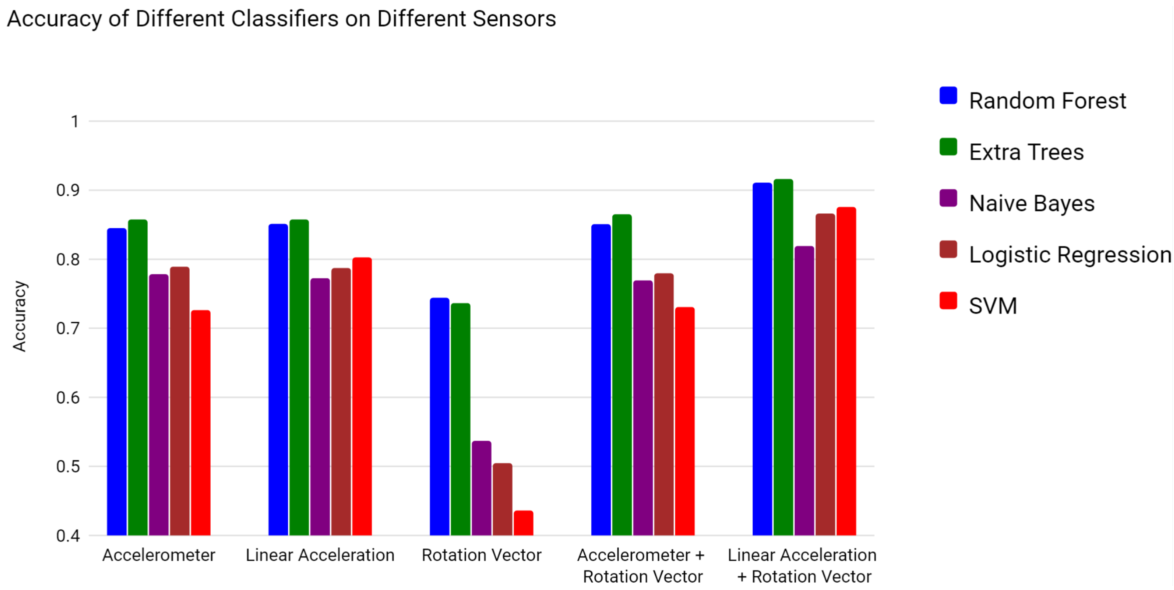


Figure 4.1: Accuracy of Different Classifiers on Different Sets of Sensors

ceptively high accuracies. Figure 4.1 depicts the accuracy of the classifiers on each set of sensors. As it can be seen, using linear acceleration sensor produces slightly better results compared to using accelerometer sensor. Using rotation vector sensor on its own does not produce great results, but in in combination with linear acceleration, these sensors provided the highest accuracy. Combining the data from the rotation vector sensor with accelerometer data did not provide much enhancement. This can be due to the fact that the orientation of device has already impacted the readings of the accelerometer sensor. From Figure 4.1 we can also see that decision tree based classifiers outperform other classifiers, regardless of the sensors being used. Among them, Extra Trees classifier provided slightly better results. Extra Trees classifier is presented in [12]. For this classifier we used 1000 estimators and we split a leaf only if it contains at least 5 samples.

Another important point about selecting the sensors for activity recognition is how well each set of sensors can distinguish different activities. Figure 4.2 depicts the performance of the Extra Trees classifier on different sets of sensors to predict each activity. To measure the quality of classification, we reported the F1-score for each label. This figure shows that using the linear acceleration sensor can accurately identify

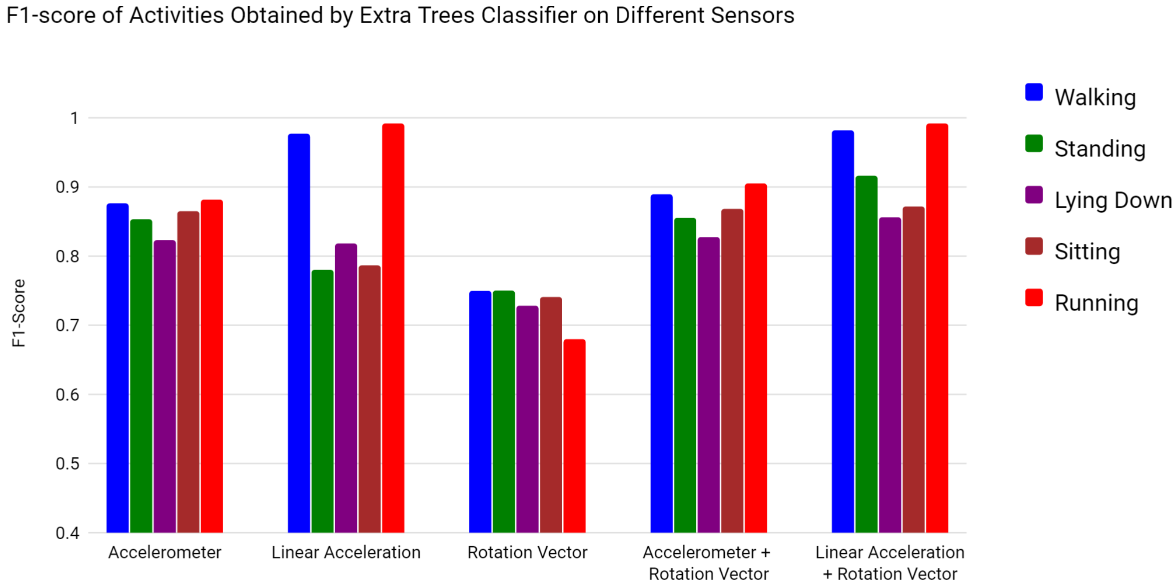


Figure 4.2: F1-Score of Activities Obtained by Extra Trees Classifier on Different Sensor Sets

walking and running activities, but this sensor works poorly for detecting standing and sitting activities. This may be due to the fact that by having access to only linear acceleration sensor, the algorithm has no sense of the orientation of the coordinate system of the device. Using the combination of linear acceleration and rotation vector sensors, our classifiers identify all activities with better or equal scores, compared to any other combination of sensor. This combination especially works better for the standing activity.

Table 4.1 shows the F1-scores obtained by different classifiers for all activities, trained on the data collected from the combination of linear acceleration and rotation vector sensors. Extra Trees classifier has the best score for all activities, except for walking, for which SVM classifier's score is slightly better. Table 4.2 shows the confusion matrix for Extra Trees classifier. The overall accuracy is 91.9%.

| | F-Score | | | | |
|---|---|---|---|---|---|
| | Random Forest | Extra Trees | Naive Bayes | Logistic Regression | SVM |
| Walking | 0.9785 | 0.9829 | 0.9693 | 0.9768 | 0.9842 |
| Standing | 0.9104 | 0.9172 | 0.7205 | 0.8753 | 0.8753 |
| Lying Down | 0.8526 | 0.8571 | 0.7201 | 0.7707 | 0.7871 |
| Sitting | 0.8647 | 0.8727 | 0.7942 | 0.7985 | 0.8066 |
| Running | 0.9928 | 0.9928 | 0.9787 | 0.9566 | 0.9913 |

Table 4.1: F1-Score Results of Different Classifiers

| | | Predicted | | | | |
|---|---|---|---|---|---|---|
| | | Walking | Standing | Lying Down | Sitting | Running |
| Actual | Walking | 690 | 4 | 0 | 1 | 1 |
| | Standing | 6 | 648 | 27 | 15 | 0 |
| | Lying Down | 2 | 40 | 591 | 63 | 0 |
| | Sitting | 7 | 25 | 65 | 600 | 0 |
| | Running | 4 | 0 | 0 | 0 | 344 |

Table 4.2: Confusion Matrix for Extra Trees Classifier

In the remainder of the thesis we use the Extra Trees classifier trained on the combination of linear acceleration and rotation vector sensors data as a baseline for the performance of the supervised methods. Table 4.3 shows the F1-scores obtained by this classifier for each subject. Although in general our classifier did pretty well in classifying the activities, it has high variance among subjects, like up to 0.11 for the sitting activity. In chapter 5 we will enhance this method by adopting it to subjects' personal activities using active learning. We will show that the interactive algorithm can reduce the mentioned variance among the users, while increasing the overall accuracy.

| Subject ID | Accuracy | F1-score | | | | |
|---|---|---|---|---|---|---|
| | | Walking | Standing | Lying Down | Sitting | Running |
| 1 | 0.985 | 1.000 | 0.973 | 0.983 | 0.975 | 1.000 |
| 2 | 0.893 | 1.000 | 0.926 | 0.836 | 0.755 | 1.000 |
| 3 | 0.946 | 0.983 | 0.957 | 0.893 | 0.924 | 1.000 |
| 4 | 0.935 | 0.991 | 0.924 | 0.865 | 0.923 | 1.000 |
| 5 | 0.843 | 0.991 | 0.905 | 0.649 | 0.739 | 0.983 |
| 6 | 0.954 | 0.983 | 0.991 | 0.924 | 0.901 | 0.982 |
| 7 | 0.950 | 0.967 | 0.957 | 0.919 | 0.949 | 0.964 |
| 8 | 0.943 | 0.973 | 0.918 | 0.897 | 0.951 | 1.000 |
| 9 | 0.839 | 1.000 | 0.818 | 0.725 | 0.642 | 1.000 |
| 10 | 0.897 | 0.949 | 0.921 | 0.800 | 0.860 | 0.982 |
| 11 | 0.858 | 0.959 | 0.708 | 0.812 | 0.862 | 1.000 |
| 12 | 0.981 | 1.000 | 0.991 | 0.966 | 0.956 | 1.000 |
| Mean | 0.919 | 0.983 | 0.916 | 0.856 | 0.870 | 0.993 |
| Std | 0.051 | 0.018 | 0.080 | 0.098 | 0.105 | 0.012 |

Table 4.3: Results Obtained by Extra Trees Classifier for Each Subject

# CHAPTER 5

## Enhancing the Classifier Using Active Learning

Active learning is a subfield of machine learning in which the learning algorithm has the ability to query an oracle to annotate some unlabeled data points. It is specially useful when the data is abundant but labeling the data is costly [29]. Active learning can be applied in two settings: stream based setting [9] in which the algorithm receives unlabeled data points one at a time and has to decide whether or not to issue a query, and pool based setting [22], in which the algorithm has access to a small set of labeled data and a large set of unlabeled data and it can select an unlabeled data point to be annotated by the oracle at each step. Our task of activity recognition fits within the stream based setting perfectly. The sensor data is being collected by the smartwatch all the time, but in order to label the data we should either ask the subjects to follow specific protocols or monitor their activities by some external resources. On the other hand, querying the user for annotating his current activity can be done easily using a smartwatch.

The main step in an active learning algorithm is to decide when to query the user, i.e. the querying strategy [29]. In this chapter we will study the performance of two popular querying strategies: Uncertainty Sampling [17] and Query by Committee [30]. In the first strategy the classifier requests the label of data points for which it has the least certainty. In the second strategy, the algorithm uses multiple models to predict the results and issues a query based on the degree of disagreement between the models.

## 5.1 Uncertainty Sampling Strategy

Since our task contains more than two labels, we used a variant of uncertainty sampling strategy defined for multi-label tasks, called margin sampling [28]. In this strategy, the algorithm issues queries for predictions which don't provide reasonable margin between the two most probable labels. For each prediction of the Extra Trees classifier, we define the probability of a label as the number of estimators which predicted the label, divided by total number of estimators in our classifier. We also define the certainty of the prediction as follows:

$$p(y|x) = \frac{\#\text{ estimators predicted } y \text{ for input } x}{\#\text{ estimators}}$$

$$certainty(x) = p(y_1|x) - p(y_2|x)$$

where $y_1$ and $y_2$ are the first and second most probable labels for $x$.

Figure 5.1 shows the cumulative distribution of certainty among correctly and incorrectly classified instances of each activity, that is, for each level of certainty, the chart shows the percentage of correctly and incorrectly classified instances with less or equal certainty. As it can be seen, in general the classifier had less certainty for misclassified instances and hence, by using this strategy our algorithm can gain information about the points on the decision boundary. Another important point that can be inferred from Figure 5.1 is that for some activities, the amount of certainty is usually less than the others, regardless of whether or not the instance is correctly classified. For example a data point which is predicted with label walking with certainty 0.6 is a good candidate to be annotated by the user, since less than 10% of the data points correctly classified as walking are obtained with certainty less than 0.6. On the other hand around 40% of the data points which are correctly classified as lying down are reported with certainty less than 0.6. Thus, comparing the certainties without considering the predicted label is not a good measure for issuing the queries because it would excessively skew the query issuance toward some activities. To avoid this problem, for each activity we define a separate threshold. We set these thresholds


Figure 5.1: Prediction Certainty for Different Activities

in such a way that they cover a fixed percentage of misclassified data instances of the activity. For example to cover 80% of misclassified instances, we get certainty threshold of 0.61 for walking and 0.42 for standing. If our algorithm predicts an activity for a data point with certainty less than the activity's threshold, the algorithm will ask the subject for the real activity being performed.

To evaluate the performance of our active learner, we use the same data set that we gathered in chapter 2. We divide each subject's data in two parts, each containing 2.5 minutes of running and 5 minutes of other activities. Let these sets be $A$ and $B$. In the cross-validation described in chapter 3, we add a querying phase for each

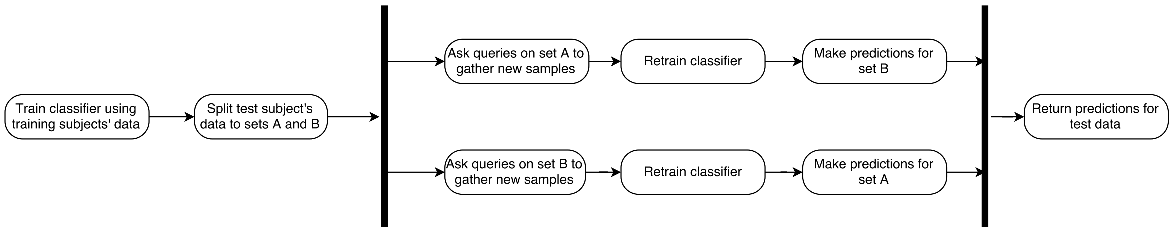


Figure 5.2: Active Learner's Work Flow

subject. In this phase we treat set $A$ as a time period in which the algorithm can issue queries and we simulate the answers of the subject with our knowledge about the real labels that we have for set $A$. After this phase, we add the data obtained by queries to the training data that algorithm already had and re-train our classifier. We add the queried samples with weight 5, same as the minimum number of samples required to split a leaf, to insure that in case of a test point falling within a tree leaf containing these points, our algorithm credits user annotated data more than combination of the default training samples in the leaf. After training the new classifier, we will test it on set $B$ to obtain predictions for data points in this set. Afterwards we repeat our algorithm by considering set $B$ as our querying time period and set $A$ as our test set to obtain predictions for all points in our dataset. Figure 5.2 shows the process in one iteration of our LOSO cross-validation. Notice that in this process, annotating the activity of a time window doesn't affect the classification of itself or time windows close to it, because they will all reside in the same subset.

Figure 5.3 shows the results of our active learner algorithm for different subjects and activities. The first point that can be inferred from this figure is that only the three worst predicted subjects had considerable increase in the accuracy. For other subjects, our classifier could achieve at least 89% accuracy without issuing any query and issuing queries did not have negative effect on them. Hence, we can safely use the active learner to reduce the variance among users and detect the outlier subjects. The maximum reduction in the accuracies was 0.04%, while we achieved and increase as big as 7.3%. With threshold set to 100%, our algorithm issued 1446 queries in total, which is equal to 46% of all data instances. Another point worth mentioning is that the

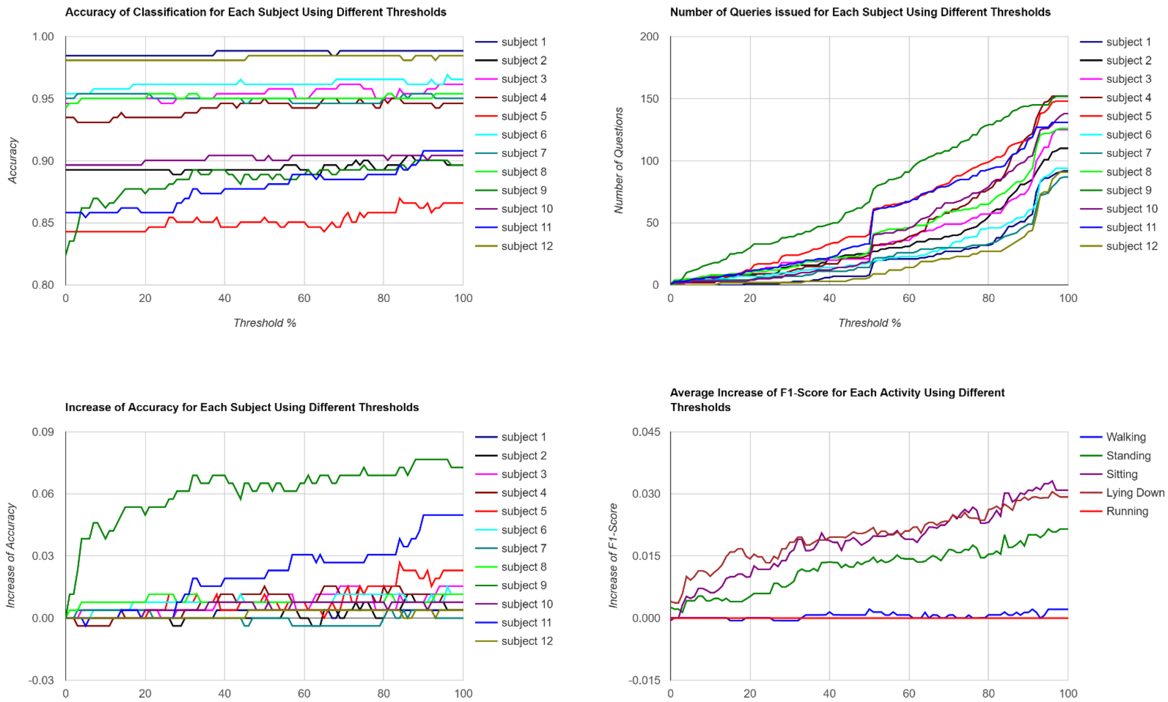


Figure 5.3: Simulation of Active Learning with Uncertainty Sampling Using Different Thresholds

number of queries issued for each subjects is correlated with the amount of increase in the accuracy that our active learner could achieve for them. For example, with threshold set to 60%, our algorithm issued only 21 queries for subject 1, which had the most accurate predictions, while it issued 91 queries for subject 9, which had the least accurate predictions. We also see that our algorithm improved the predictions for standing, sitting and lying down activities, but it did not have noticeable impact on walking and running activities. Table 4.1 shows that walking and running were predicted correctly more than 98% of the times and hence, active learner did not have much room for improvement.

Table 5.1 shows the accuracy of the classifier and F1-scores for different activities for all subjects with threshold set to 90%. At this threshold, the accuracy gets close to its highest value for most of the subjects while the number of issued queries, 1059, is equal to 34% of all data instances. In this table, the amount of increase for all rows, including the last two, refers to the amount of increase in the values compared to Table 4.3. Using this querying strategy, we could achieve around 3% increase in F1-scores for

| Subject ID | New Value | | | | | | Amount of Increase | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | F1-Score | | | | | Accuracy | F1-Score | | | | |
| | | Walking | Standing | Lying Down | Sitting | Running | | Walking | Standing | Lying Down | Sitting | Running |
| 1 | 0.989 | 1.000 | 0.973 | 0.991 | 0.983 | 1.000 | 0.004 | 0.000 | 0.000 | 0.009 | 0.008 | 0.000 |
| 2 | 0.900 | 1.000 | 0.927 | 0.846 | 0.778 | 1.000 | 0.008 | 0.000 | 0.001 | 0.010 | 0.023 | 0.000 |
| 3 | 0.954 | 0.991 | 0.966 | 0.911 | 0.924 | 1.000 | 0.008 | 0.008 | 0.009 | 0.018 | 0.000 | 0.000 |
| 4 | 0.946 | 1.000 | 0.933 | 0.889 | 0.933 | 1.000 | 0.011 | 0.009 | 0.009 | 0.024 | 0.010 | 0.000 |
| 5 | 0.862 | 0.974 | 0.896 | 0.719 | 0.789 | 0.983 | 0.019 | -0.017 | -0.009 | 0.071 | 0.050 | 0.000 |
| 6 | 0.962 | 0.983 | 0.991 | 0.940 | 0.920 | 0.982 | 0.008 | 0.000 | 0.000 | 0.016 | 0.019 | 0.000 |
| 7 | 0.954 | 0.967 | 0.957 | 0.929 | 0.957 | 0.964 | 0.004 | 0.000 | 0.000 | 0.010 | 0.008 | 0.000 |
| 8 | 0.950 | 0.982 | 0.933 | 0.907 | 0.951 | 1.000 | 0.008 | 0.009 | 0.015 | 0.010 | 0.000 | 0.000 |
| 9 | 0.900 | 1.000 | 0.911 | 0.821 | 0.814 | 1.000 | 0.077 | 0.000 | 0.093 | 0.096 | 0.172 | 0.000 |
| 10 | 0.904 | 0.949 | 0.935 | 0.824 | 0.860 | 0.982 | 0.008 | 0.000 | 0.015 | 0.024 | 0.000 | 0.000 |
| 11 | 0.908 | 0.967 | 0.808 | 0.896 | 0.904 | 1.000 | 0.050 | 0.008 | 0.099 | 0.084 | 0.042 | 0.000 |
| 12 | 0.985 | 1.000 | 1.000 | 0.966 | 0.965 | 1.000 | 0.004 | 0.000 | 0.009 | 0.000 | 0.009 | 0.000 |
| Mean | 0.935 | 0.984 | 0.936 | 0.887 | 0.898 | 0.993 | 0.016 | 0.001 | 0.020 | 0.031 | 0.029 | 0.000 |
| Std | 0.039 | 0.017 | 0.051 | 0.074 | 0.071 | 0.012 | -0.013 | 0.000 | -0.029 | -0.024 | -0.034 | 0.000 |

Table 5.1: Results Obtained by Active Learner with Threshold Set to 90%

sitting and lying down and 2% increase for standing. In addition, standard deviation for sitting, standing and lying down are reduced by 0.034, 0.029, and 0.024 which makes the standard deviations relatively 36%, 24%, and 32% smaller. In addition, this table shows that we can achieve up to 17.2% increase in F1-scores for sitting, 9.9% increase for standing and 9.6% increase for lying down, without reduction in almost any of the scores.

## 5.2 Query by Committee Strategy

Query by Committee strategy issues queries based on the level of disagreement among different models for predicting the results. To implement this strategy for our task, we used three different models: Extra Trees Classifier, SVM with linear kernel and Naive Bayes classifier. We issue a query when any pair of these classifiers predict different labels for a data point. After issuing the queries, we re-train the Extra Trees classifier and use this model for returning the final decision of the algorithm. To measure the performance of this strategy, we used LOSO cross-validation with the same procedure as Section 5.1, depicted in Figure 5.2. Our algorithm issued 566 queries, running on

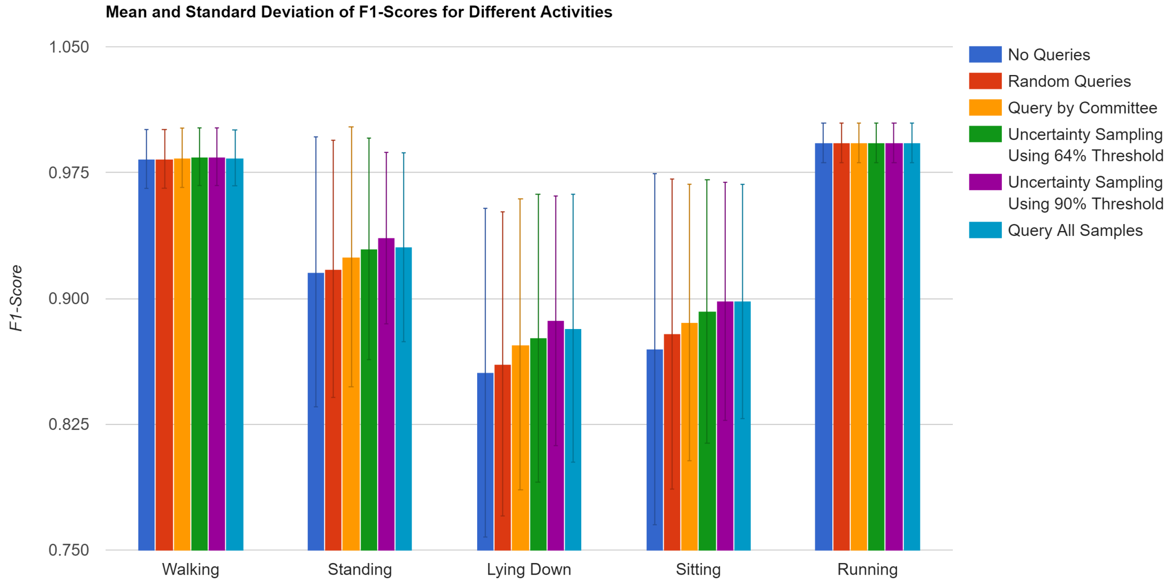


Figure 5.4: Comparison of the Performance of Different Querying Strategies

our data set. This number of queries is equal to the number queries issued by the uncertainty sampling strategy with threshold set to 64%. Figure 5.4 shows the F1-scores and standard deviations of both strategies. In order to assess the performance of both strategies, we also simulated an active learner which issues queries uniformly with probability 0.181 to obtain a similar number of queries. It can be seen that both strategies outperform randomly issuing queries. Also using a same number of queries, uncertainty sampling strategy slightly performs better than the query by committee strategy.

Uncertainty sampling is also more flexible for our task, since we can modify the query issuance rate by modifying the algorithms threshold. As it can be seen in Figure 5.4, using 90% threshold for uncertainty sampling strategy completely outperforms other strategies, even issuing queries for all instances.

# CHAPTER 6

# Related Work

Human activity recognition using accelerometer data has been vastly studied on different kinds of activities and different placement configurations on human body. A lot of these studies use multiple accelerometers on different body locations [8]. Although using multiple accelerometers enables the system to achieve more information about the movements of the subjects and hence, recognize more kinds of activities, it makes the system harder to use, especially for remote health monitoring. In this chapter we first review the works involving supervised machine learning algorithms for activity recognition, based on a single accelerometer placed on the wrist. Afterwards we discuss the efforts been done on using active learning for human activity recognition.

Chernbumroong et al. [7] investigated the use of an accelerometer embedded in a wrist worn sports watch for activity classification. They analyzed the same set of activities we selected for our study. Their data is collected from 7 participants and the total length of data is 35 minutes which is on average 1 minute of data per activity per subject. Authors compared the performance of Decision Tree C4.5 and Artificial Neural Network algorithms using four different feature sets and similar to us, they concluded that the decision tree algorithm performs better using any feature set, with 94.13% accuracy in the best case. The accuracies are obtained by 5-fold cross-validation.

Yang [36] has studied activity recognition using an accelerometer placed on the wrist which is quite similar to using a smartwatch. He collected data for 6 activities: sitting, standing, walking, running, driving, and bicycling. He reported 90.6% accuracy for classifying the activities using C4.5 Decision Tree classifier. This accuracy is obtained by training the classifier on data collected from 3 subjects and testing on the data collected from the fourth subject, which calls the generalizability of the results into question, since, as it can be seen in Table 4.3, the accuracy of our method ranges between 83.9% and 98.5% among different subjects.

Cleland et al. [8] studied optimal placement of accelerometers for detecting everyday activities. They reported that using a single accelerometer on the hip achieves highest accuracy, 97.81%. They analyzed 7 activities: lying down, running, sitting, ascending the stairs, descending the stairs, standing, and walking. For a single accelerometer placed on the wrist they reported 95.88% accuracy, using SVM classifier. The study has been performed on 8 subjects and the results are reported based on 10-fold cross-validation.

Olguın & Pentland [25] studied 7 configurations of accelerometer placement on human body to detect 8 activities: sitting, running, squatting, standing, crawling, lying down and moving hand while standing. They used hidden Markov models as the classification method. For a single accelerometer placed on the wrist, they reported 55.45% accuracy and they could achieve 92.13% accuracy by using 3 accelerometer placed on the wrist, hip, and chest. The study has been done on 3 subjects. The results are obtained using 9-fold cross-validation.

Maurer et al. [21] have explored activity monitoring using a multi sensor platform worn on different body positions. They analyzed the data generated by a biaxial accelerometer and a light sensor. The study analyzes 6 activities: running, sitting, standing, walking, ascending the stairs, and descending the stairs. Authors reported 87.1% accuracy for the wrist placement of their device. They also reported that the best placement is on subject's shirt, resulting in 89.5% accuracy. Six subjects have participated in the study and the results are obtained by 5-fold cross validation using decision tree algorithm.

Mannini et al. [19] performed a comprehensive study on 26 daily activities, over the data gathered from 33 subjects. They clustered the activities in 4 categories: ambulation, cycling, sedentary, and other activities. The study investigates the performance of different machine learning models to detect activities in each cluster based on LOSO cross-validation. Using SVM classifier they could identify the activity categories with 84.7% accuracy, using data from a wrist worn accelerometer. Authors also reported that using an accelerometer on the ankle, they could achieve 95.0% accuracy. They also explored the performance of their classifier for different feature extraction time windows of 2, 4 and 12.8 seconds. They reported that the best result was obtained using 12.8 seconds time interval which is close to our interval length of 10 seconds . This study also confirms that 10-fold cross-validation for activity recognition produces less generalizable results compared to LOSO cross-validation.

Guiry et al. [13] investigated the performance of both smartwatches and smartphones for activity recognition. Their study involves 10 subjects performing 9 activities: walking, running, cycling, standing, sitting, elevator ascend, elevator descend, stair ascend, and stair descend. Authors analyzed the performance of different supervised machine learning methods on two datasets: a balanced dataset in which each activity has same number of instances (similar to our dataset) and an unbalanced dataset which resembles the activities done by humans in a real world scenario. In the latter case, the the overall true positive rates will be skewed towards the activities with greater share in the dataset. By only using the smartwatch, authors could achieve 56.89% accuracy on the balanced dataset and 89.26% on the unbalanced dataset, though it is not mentioned how these accuracies are computed. The smartwatch used in this study only provided accelerometer sensor and other motion sensors like gyroscope and magnetometer were not available. Using cellphone accelerometer the authors achieved 75.00% and 95.50% accuracy for balanced and unbalanced datasets respectively. For the unbalanced dataset, authors could increase the accuracy slightly by fusing additional cellphone sensors including magnetometer, gyroscope and pressure sensor, obtaining 94.60% accuracy. They reported that decision tree classifier had the best performance for smartwatch based classification, while SVM performed better in the smartphone based task.

| Reference | Activities | # Participants | Accuracy Calculation Method | Classification Method | Accuracy |
|---|---|---|---|---|---|
| This study | Walking, Standing, Sitting, Lying Down, Running | 12 | LOSO cross-validation | Decision Tree | 91.9% |
| Chernbumroong [7] | Walking, Standing, Sitting, Lying Down, Running | 7 | 5-fold cross-validation | Decision Tree | 94.13% |
| Yang [36] | sitting, standing, walking, running, driving, bicycling | 3 | train on 3 subjects, test on 1 subject | Decision Tree | 90.6% |
| Cleland [8] | lying down, running, sitting, acceding stairs, descending stairs, standing, walking | 8 | 10-fold cross-validation | SVM | 95.88% |
| Olgun [25] | sitting, running, squatting, standing, crawling, lying down, moving hands while standing | 3 | 9-fold cross-validation | HMM | 55.45% |
| Maurer [21] | running, sitting, standing, walking, ascending and descending the stairs | 6 | 5-fold cross-validation | Decision Tree | 87.1% |
| Mannini [19] | ambulation, sedentary, cycling, other activities | 33 | LOSO cross-validation | SVM | 84.7% |
| Guiry [13] | walking, running, cycling, standing, sitting, elevator ascend, elevator descend, stair ascend, stair descend | 10 | NA | Decision Tree | 56.89% (balanced) 89.26% (unbalanced) |

Table 6.1: Review of Studies on Activity Recognition Using Supervised Machine Learning Techniques Based on a Wrist Worn Accelerometer

Table 6.1 summarizes the reviewed literature on supervised machine learning activity recognition using a wrist worn accelerometer. In regard to studies related to active learning for activity recognition, Longstaff et al. [18] studied multiple semi-supervised algorithms for activity recognition. Settings of their study was defined to identify three activities: walking, running, and staying in one place, using accelerometer data and GPS speed recorded by a smartphone. Authors analyzed active learning, self-learning, En-Co-Training and democratic co-learning for this task and concluded that active learning achieves the best results. They also reported that active learning performs well when the classifiers accuracy is low, but it does not hurt the accuracy in any case. This claim is also supported by our study.

Stikic et al. [31] focused on using active learning to reduce the number of data instances required for training a classification model by selecting the most informative data points to be labeled. Their study was based on data recorded from a smart home environment which had been equipped with multiple infrared sensors in different places to keep track of subjects' location. Also each subject wore 3 tri-axial accelerometers on the wrist, hip, and thigh. Authors showed that active learning can achieve similar or sometimes better performance compared to supervised methods, with significantly less training data.

Abdallah et al. [2] used active learning to obtain subject adopted classification models to detect activities of different users. The dataset used in this study is also based on a multi-sensor enviroment. Authors proposed a cluster based active learning method that issues queries on data points far from cluster centers. They concluded that their proposed method using active learning shows improved performance over different supervised methods.

# CHAPTER 7

## Conclusion

In this thesis, we have presented a smartwatch-based system to detect five daily activities: walking, sitting, standing, lying down, and running. Twelve subjects were asked to perform these activities, yielding a total of 540 minutes of data. An analysis of different sets of smartwatch sensors demonstrated that using Android's software-based sensors data, namely linear acceleration sensor and rotation vector sensor, outperformed physical accelerometer data, which is widely used in previous studies. We also investigated the performance of different machine learning algorithms, among which the Extra Trees classifier achieved the highest accuracy of 91.9%. Furthermore, we explored the performance of active learning for obtaining user specific models using two different strategies: uncertainty sampling and query by committee. Our study showed that active learning can increase accuracy for subjects with less than 89% accuracy, without reducing accuracy for other subjects. Among the aforementioned strategies, uncertainty sampling generated better models which improved classifier performance in both accuracy and standard deviation. Uncertainty sampling with a 34% query rate yielded an improvement in the F1-scores of "sitting" and "lying down", and "standing" by 0.3, 0.3, and 0.2 respectively, while also reducing the standard deviations of the F1-scores of the aforementioned activities by 36%, 32% and 24% of their previous values.

# References

[1] Android Motion Sensors developer guide. `https://developer.android.com/guide/topics/sensors/sensors_motion.html`. Accessed: 2016-05-11.

[2] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy. Streamar: incremental and active learning with evolving sensory data for activity recognition. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 1163–1170. IEEE, 2012.

[3] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[4] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pages 1–10. VDE, 2010.

[5] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.

[6] J. Bussmann, W. Martens, J. Tulen, F. Schasfoort, H. Van Den Berg-Emons, and H. Stam. Measuring daily behavior using ambulatory accelerometry: the activity monitor. *Behavior Research Methods, Instruments, & Computers*, 33(3):349–356, 2001.

[7] S. Chernbumroong, A. S. Atkins, and H. Yu. Activity classification using a single wrist-worn accelerometer. In *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on*, pages 1–6. IEEE, 2011.

[8] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7):9183–9200, 2013.

[9] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[10] J. Dooley and J. M. McCarthy. Spatial rigid body dynamics using dual quaternion components. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 90–95. IEEE, 1991.

[11] P. Geurts. Pattern extraction for time series classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer, 2001.

[12] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[13] J. J. Guiry, P. van de Ven, and J. Nelson. Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices. *Sensors*, 14(3):5687–5701, 2014.

[14] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim. Activity recognition using wearable sensors for elder care. In *2008 Second International Conference on Future Generation Communication and Networking*, volume 2, pages 302–305. IEEE, 2008.

[15] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[16] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[17] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[18] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–7. IEEE, 2010.

[19] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell. Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in sports and exercise*, 45(11):2193, 2013.

[20] M. Mathie, J. Basilakis, and B. Celler. A system for monitoring posture and physical activity using accelerometers. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd annual international conference of the IEEE*, volume 4, pages 3654–3657. IEEE, 2001.

[21] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4–pp. IEEE, 2006.

[22] A. K. McCallumzy and K. Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

[23] B. Mortazavi, E. Nemati, K. VanderWall, H. G. Flores-Rodriguez, J. Y. J. Cai, J. Lucier, A. Naeim, and M. Sarrafzadeh. Can smartwatches replace smartphones for posture tracking? *Sensors*, 15(10):26783–26800, 2015.

[24] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[25] D. O. Olguın and A. S. Pentland. Human activity recognition: Accuracy across common locations for wearable sensors. In *Proceedings of 2006 10th IEEE International Symposium on Wearable Computers, Montreux, Switzerland*, pages 11–14. Citeseer, 2006.

[26] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of neuroengineering and rehabilitation*, 9(1):1, 2012.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[29] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[30] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[31] M. Stikic, K. Van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 81–88. IEEE, 2008.

[32] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015.

[33] E. M. Tapia, S. S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, pages 158–175. Springer, 2004.

[34] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *ICML*, pages 406–414. Citeseer, 1999.

[35] G. M. Weiss and H. Hirsh. Learning to predict rare events in event sequences. In *KDD*, pages 359–363, 1998.

[36] J. Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pages 1–10. ACM, 2009.