AN ABSTRACT OF THE THESIS OF

Yonglei Zheng for the degree of Master of Science in Computer Science presented on August 17, 2012.

Title: Predicting Activity Type from Accelerometer Data

Abstract approved: _____

Weng-Keen Wong

The study of physical activity is important in improving people's health as it can help people understand the relationship between physical activity and health. Accelerometers, due to its small size, low cost, convenience and its ability to provide objective information about the frequency, intensity, and duration of physical activity, has become the method of choice in measuring physical activity. Machine learning algorithms based on the featurized representation of accelerometer data have become the most widely used approaches in physical activity prediction. To improve the classification accuracy, this thesis first explored the impact of the choice of data (raw vs processed) as well as the choice of features on the performance of various classifiers. The empirical results showed that the machine learning algorithms with strong regularization capabilities always performed better if provided with the most comprehensive feature set extracted from raw accelerometer signal.

Based on the hypothesis that for some time series, the most discriminative information could be found at subwindows of various sizes, the Subwindow Ensemble Model (SWEM) was proposed. The SWEM was designed for the accelerometer-based physical activity data, and classified the time series based on the features extracted from subwindows. It was evaluated on six time series datasets. Three of them were accelerometer-based physical activity data, which the SWEM was designed for, and the rest were different types of time series data chosen from other domains. The empirical results indicated a strong advantage of the SWEM over baseline models on the accelerometer-based physical activity data. Further analysis confirmed the hypothesis that the most

discriminative features could be extracted from subwindows of different sizes, and they were effectively used by the SWEM.

Predicting Activity Type from Accelerometer Data

by

Yonglei Zheng

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 17, 2012
Commencement June 2013

Master of Science thesis of Yonglei Zheng presented on August 17, 2012.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Yonglei Zheng, Author

ACKNOWLEDGEMENTS

I would like to thank my entire committee, Weng-Keen Wong, Stewart Trost, Alan Fern and Kenneth H. Funk II, for their support and constructive comments. I am sincerely and heartily grateful to my advisor, Weng-Keen Wong, for the support and guidance he showed me throughout my research and thesis writing. This thesis would not have been possible without his help. I am grateful for Stewart Trost for contributing his domain knowledge and expertise regarding physical activity. I would like to thank my parents for their support and encouragement throughout my graduate studies. Finally, I offer my regards and blessings to all of those who supported me in any respect during the completion of the program.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

1.1 Introduction

Accurate measurement of physical activity is essential for performing physical activity surveillance, understanding the relationship between physical activity dose and health outcomes, identifying the influence of physical activity, detecting people at risk, and evaluating the effectiveness of intervention strategies designed to increase physical activity [11, 50, 42]. Self-reports have been the traditional approaches of providing information about the physical activity people engage in [11]. However, they are susceptible to subjective factors, such as recall bias and social desirability, thus lacking accuracy. For instance, self-reports tend to overestimate the time spent in unstructured daily physical activities, such as walking [11, 2, 37, 46]. There are many methods available other than self-reports, such as double-labeled water, direct observation, calorimetry, HR monitors, and accelerometers [11, 50]. Among all these methods, accelerometers are the most promising alternatives to self-reports. In contrast to self-reports, accelerometers are immune to subjective influences. They also provide objective information about the frequency, intensity, and duration of physical activity, and therefore, have become the method of choice in measuring physical activity [11].

One of the weaknesses of the accelerometer data is that it cannot be directly translated to the type of activity people engage in. Recently, machine learning approaches, such as quadratic discriminant analysis [34], decision trees [6], and artificial neural networks [38, 42], have been explored to cope with such weakness of accelerometers.

Since machine learning approaches are accepted as promising alternatives to traditional physical activity assessment technologies, our study focused on the methods for improving the classification accuracy of physical activity with such approaches. We were interested in the choice of accelerometer data, the features and the design of machine learning models that resulted in improved classification accuracies. Based on our observation of the accelerometer-based physical activity data, a hypothesis was suggested that the most useful features might exist in subwindows of various sizes for such data.

The hypothesis led to the design of the Subwindow Ensemble Model (SWEM). Empirical evidence has been found to support our hypothesis, and the SWEM also resulted in better performance than classic approaches.

1.2 Related Work

Various machine learning based techniques, including hidden Markov models (HMM) [34, 18], decision trees [6, 39, 3, 5], support vector machines (SVM) [43], and artificial neural networks (ANN) [38, 42, 11, 10, 45], have been applied to physical activity recognition. Among these approaches, ANNs are the most popular ones. Apart from these methods, there are numerous approaches proposed by people from different domains that can potentially work for physical activity prediction.

The k-nearest neighbor (k-NN) algorithm is very popular in time series classification. Various distance measures can be used by k-NN. Although being extremely simple, Euclidean distance has been shown to be surprisingly competitive in terms of accuracy [22]. Dynamic time warping (DTW) [4, 23] handles distortions in time series better than pure Euclidean distance. It is well known in automatic speech recognition to cope with different speaking speeds. Combined with k-NN, it has also achieved hard-to-beat performance [49]. Keogh et al. proposed a symbolic representation of time series (SAX) [26], making an enormous wealth of existing symbolic approaches available for time series applications.

Time series shapelets [51] was introduced as a primitive based approach for time series data mining. Briefly speaking, shapelets are local patterns in time series that are highly predictive of a class, and are thus very discriminative features for building classifiers, such as decision trees, as well as certain visualization and summarization tasks [31]. The logical-shapelets algorithm upgraded the original shapelets algorithm by introducing techniques to speedup search for shapelets, and creating more expressive shapelets with logical operations [31]. Another primitive based model uses the bag-of-features approach [52], which is widely used in computer vision.

Chapter 2: Predicting Physical Activity with Feature-Based Approaches

2.1 Introduction

The machine learning algorithms based on the featurized representation of the accelerometers data have become the most widely used approaches in physical activity prediction. In this section, a set of experiments were conducted on a specific physical activity dataset to study the impact of the choice of data (raw vs processed) and features on the performance of various classifiers. Our goal was to develop an informative feature representation of the data and determine which machine learning algorithms made the most accurate predictions based on this representation.

Most people predict activity based on counts integrated from raw accelerometer signal. However, the results showed that machine learning models based on raw accelerometer signal always performed better than models based on counts. Three sets of features were tested in our experiments. The first set was a subset of the second set, and the second set was a subset of the third one. The results showed that more features could make predictions more accurate. However, only models with regularization could benefit from such a large feature set.

2.2 Experiments

2.2.1 Data Collection

The physical activity data used in our experiments were collected from 52 children and adolescents (mean age 13.7 +/- 3.1 y, 28 boys, 24 girls). They completed 12 standardized activity trials in a controlled lab environment within a 2-week time period. Each trial lasted 5 minutes, except for the lying down trial, which lasted 10 minutes. The 12 trials were categorized into 8 distinct physical activity classes: lying down, sitting, standing, household chores, walking, running, basketball, and dancing. Figure 2.1 shows

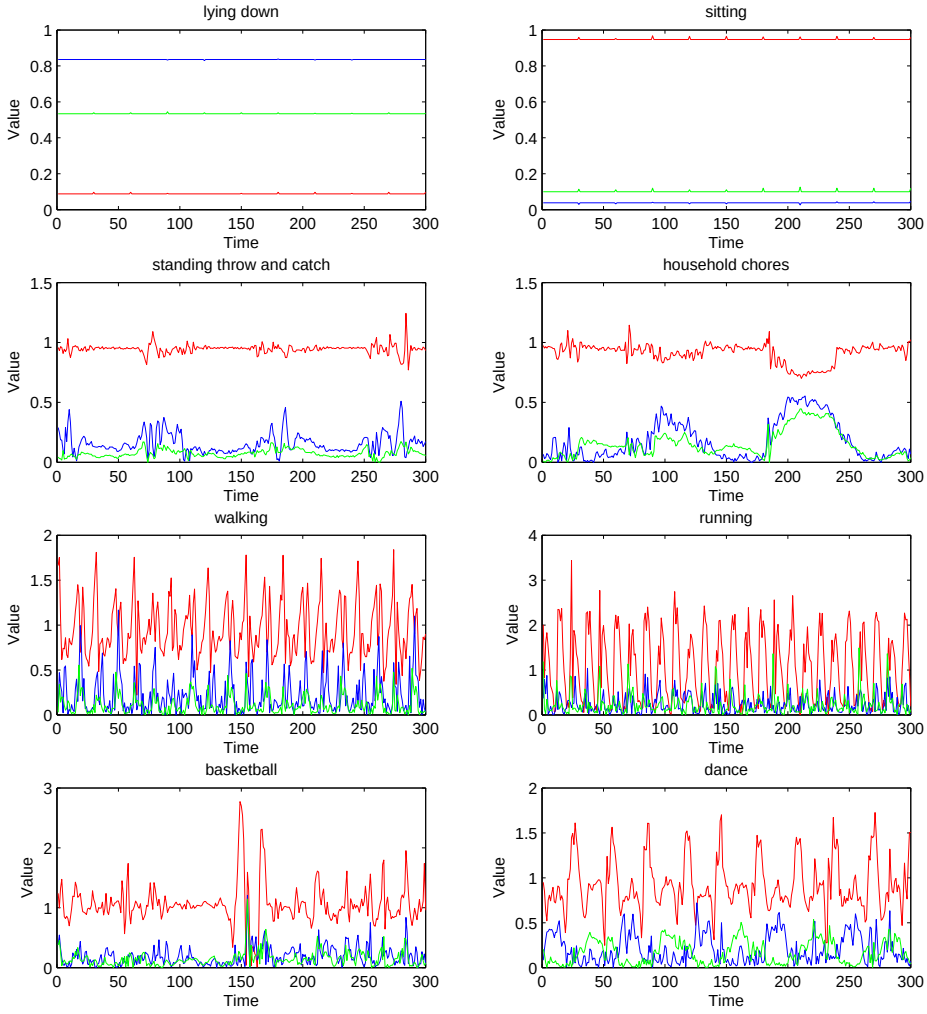

Figure 2.1: An example of time series from all eight activities. These plots are from triaxial accelerometer data collected at 30 Hz, and each color in a plot represents one axis. Each time series instance lasts for 10 seconds.

an example of 8 activities. A complete description of the activities trials can be found in [45].

The activity data were recorded by an ActiGraph GT3X+ triaxial accelerometer (ActiGraph, LLC; Pensacola, FL) mounted on the subject's right hip at a sampling rate of 30 Hz. ActiGraph propriety software (ActiLife Version 5.8) was used to download the raw triaxial acceleration signal recorded over the entire lab visit. To compare with the models using processed count data, ActiLife propriety software was used to filter and process the raw acceleration data into activity counts per second.

The predictions were based on the 10-second time windows segmented from raw (30 Hz) and processed (1 Hz) acceleration signal recorded between minutes 2.5 and 4.5 of each activity trial. Table 2.1 shows the 15 features used in our experiments, which were chosen from an extensive list of time domain features described by Liu et al. [27]. Feature 1-14 were for single axis, and Feature 15 was extracted from each pair of axes. The first feature set the models were tested on involved all 15 features, and is referred to as 'all features'. The models were also tested on two subsets of 'all features' – 'uniaxial Staudenmayer features', which were the same features used by Staudenmayer et al. [42] on the vertical axis, including percentiles (10th, 25th, 50th, 75th, 90th) (Feature 6) and lag one autocorrelation (Feature 8) and 'triaxial Staudenmayer features', which were Staudenmayer features extracted from the vertical, medio-lateral, and antero-posterior axes.

2.2.2 Models

Three models (an artificial neural network (ANN), logistic regression and adaboost) were tested in our experiments.

The ANN was chosen because it was the most widely used machine learning algorithm in physical activity prediction. A feed-forward neural networks with a single hidden layer was tested. The 'nnet' [48] package in R [36] was used as the ANN implementation in our experiment.

Logistic regression models with L1 regularization (lasso penalty) and L2 regularization (ridge penalty) were tested in our experiments. The 'glmnet' [15] package in R was used as the logistic regression implementation in our study.

Adaboost was included as an ensemble algorithm. Decision trees were used as the

Features	Time Complexity	References
1. Sum of values of a period of time: $\sum_{i=1}^{N} s_i$.	$O(N)$	[7, 9, 30, 34, 41]
2. Mean: $\mu_s = \frac{1}{N}\sum_{i=1}^{N} s_i$.	$O(N)$	[5, 6, 19, 28, 29]
3. Standard deviation: $\sigma_s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - u_s)}$.	$O(N)$	[5, 6, 19, 29, 40]
4. Coefficients of variation: $c_v = \frac{\sigma_s}{\mu_s}$.	$O(N)$	[9, 38, 40]
5. Peak-to-peak amplitude: difference between maximum and minimum signal values: $max(S) - min(S)$.	$O(\log(N))$	[5, 6]
6. Percentiles $10^{th}, 25^{th}, 50^{th}, 75^{th}, 90^{th}$.	$O(\log(N))$	[19, 29, 42]
7. Interquartile range: difference between the 75^{th} and 25^{th} percentiles.	$O(\log(N))$	[28, 38]
8. Lag-one-autocorrelation: $\frac{\sum_{i=1}^{N-1}(s_i-\mu_s)(s_{i+1}-\mu_s)}{\sum_{i=1}^{N}(s_i-\mu_s)^2}$.	$O(N)$	[42]
9. Skewness: $\frac{\frac{1}{N}\sum_{i=1}^{N}(s_i-\mu_s)^3}{(\frac{1}{N}\sum_{i=1}^{N}(s_i-\mu_s)^2)^{\frac{3}{2}}}$, measure of asymmetry of the signal probability distribution.	$O(N)$	[38]
10. Kurtosis: $\frac{\frac{1}{N}\sum_{i=1}^{N}(s_i-\mu_s)^4}{(\frac{1}{N}\sum_{i=1}^{N}(s_i-\mu_s)^2)^3} - 3$, degree of the peakedness of the signal probability distribution.	$O(N)$	[38]
11. Signal power: $\sum_{i=1}^{N} s_i^2$.	$O(N)$	[38]
12. Log-energy: $\sum_{i=1}^{N} \log(s_i^2)$.	$O(N)$	[25]
13. Peak intensity: number of signal peak appearances within a certain period of time.	$O(N)$	[38]
14. Zero crossings: number of times the signal crosses its median.	$O(N)$	[40]
15. Correlation between axes: $\frac{\sum_{i=1}^{N}(s_i-\mu_s)(v_i-\mu_v)}{\sqrt{\sum_{i=1}^{N}(s_i-\mu_s)\sum_{i=1}^{N}(v_i-\mu_v)}}$.	$O(N)$	[5, 29]

Table 2.1: Time series features. The 'Time Complexity' column shows the time complexity of the corresponding statistics assuming each time series has N data points. The 'References' column shows the papers in which the corresponding features were used.

base classifiers. The AdaboostM1 [14] and J48 decision tree classifier [35] in WEKA [17] were used in our experiments.

2.2.3 Model Training and Evaluation

Each model was developed and tested using a variation of the k-fold cross-validation procedure involving training, validation (tuning of the network parameters), and testing [45]. The dataset was randomly split by subject into 3 disjoint subsets as training, validation and testing data, and all three contained approximately the same number of subjects, and the same activity distribution. To tune a model, it was first trained with different parameters on training data, and then tested on validation data. A grid search was applied to search for the best parameters for models. The number of units in the hidden layer (1 to 30), and the weight decay (0.0, 0.5 and 1.0) parameters of ANN models were tuned. The penalty coefficient λ was tuned for logistic regression. The candidate values of λ were automatically generated by 'glmnet' during training. For adaboost, the number of iterations (5, 10, 20, 50, 100, 200 and 500) as well as the confidence threshold for pruning (1E-6, 1E-4, 1E-2, 0.1, 0.25 and 0.5) of the J48 decision tree were tuned. The parameters that resulted in the highest classification accuracy on validation data were selected, and the corresponding model was then evaluated on testing data and its accuracy was reported as the final evaluation for the model. With 10 random splits, and 6 different training-validation-testing assignments per split, each model was evaluated on a total of 60 training-validation-testing iterations.

The performance of each model was evaluated based on the average accuracy (percentage of correctly classified 10-second windows) over all 60 training-validation-testing iterations. ANOVA and t-tests were used to test the statistical significance of the performance differences between models.

2.2.4 Results

We first investigated the influence of three factors on classification accuracy:

1. Sampling frequency of accelerometers (1 Hz vs 30 Hz),
2. Features (uniaxial Staudenmayer features vs triaxial Staudenmayer features),

Algorithm	Data	Uniaxial Staudenmayer			Triaxial Staudenmayer		
		ModelID	Mean	SD	ModelID	Mean	SD
ANN	1 Hz	ANN_P1	0.6862	0.0138	ANN_P3*	0.7934	0.0139
	30 Hz	ANN_R1	0.8514	0.0176	ANN_R3	0.8519	0.0197
Glmnet+L1	1 Hz	GL1_P1	0.6724	0.0142	GL1_P3*	0.7842	0.0133
	30 Hz	GL1_R1	0.8250	0.0175	GL1_R3*	0.8647	0.0187

Table 2.2: Testing results of ANN and logistic regression with L1 regularization (Glmnet+L1) with single axis and triaxial Staudenmayer features from processed data (1 Hz) and raw acceleration signal (30 Hz). The asterisk indicates that the t-test between the triaxial model and the uniaxial model in the same row has a p-value < 0.05. All t-tests between 1 Hz and 30 Hz models with the same algorithm and features have p-values < 0.05.

3. Algorithms (artificial neural networks (ANN) vs logistic regression with L1 regularization (Glmnet+L1)).

Table 2.2 presents our results. The results of ANN and Glmnet+L1 are shown because ANN was the most widely used algorithm in physical activity prediction, and Glmnet+L1 was the best performing algorithm in our experiments. We leave out the results of the other two algorithms so that we do not clutter up the table. Triaxial models clearly outperformed uniaxial models in terms of classification accuracy. For ANN based models with 1 Hz data, adding features extracted from two additional axes improved accuracy from 0.6862 (ANN_P1) to 0.7934 (ANN_P3). However, for ANN based models with 30 Hz data, the performance difference was not significant. Both ANN_R1 (0.8514) and ANN_R3 (0.8519) showed very similar accuracies. This issue did not appear for Glmnet+L1 based models. For them, triaxial features consistently outperformed uniaxial features, by improving the accuracy from 0.6724 (GL1_P1) to 0.7842 (GL1_P3) (1 Hz), and from 0.8250 (GL1_R1) to 0.8647 (GL1_R3) (30 Hz).

Table 2.2 also shows that the models based on 30 Hz data always outperformed their counterparts based on 1 Hz data. For instance, ANN_R1 outperformed its counterpart ANN_P1 by 0.1652, while ANN_R3 outperformed ANN_P3 by 0.0585. For Glmnet+L1 models, the performance improvements were 0.1526 (uniaxial) and 0.0805 (triaxial). Combining comparisons above, the triaxial Staudenmayer features from 30 Hz data was the best choice for both ANN (ANN_R3) and Glmnet+L1 (GL1_R3) models, while the

uniaxial Staudenmayer features from 1 Hz data (ANN_P1 and GL1_P1) was the weakest.

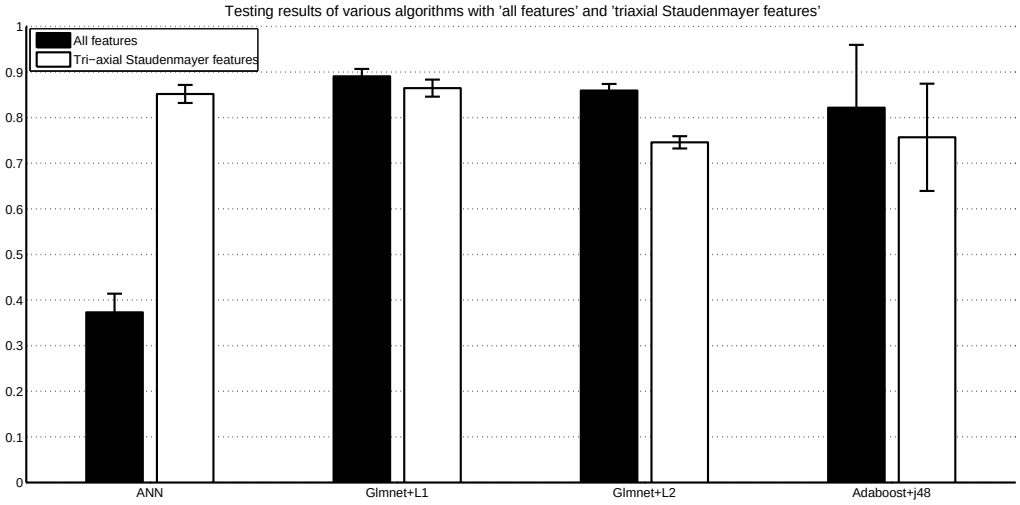

Figure 2.2: Testing results of various algorithms with 'all features' and 'triaxial' Staudenmayer features from raw acceleration signal (30 Hz)

Feature	Algorithm	Mean	SD	p-value against ANN
All features	ANN	0.3731	0.0409	–
	Glmnet+L1	0.8906	0.0161	$p < 0.0001$
	Glmnet+L2	0.8594	0.0144	$p < 0.0001$
	Adaboost+j48	0.8218	0.1375	$p < 0.0001$
Triaxial Staudenmayer	ANN	0.8519	0.0197	–
	Glmnet+L1	0.8647	0.0187	$p = 4.01E-4$
	Glmnet+L2	0.7459	0.0134	$p < 0.0001$
	Adaboost+j48	0.7568	0.1175	$p < 0.0001$

Table 2.3: Testing results of various algorithms with 'all features' and 'triaxial Staudenmayer features'

Figure 2.2 and Table 2.3 present the results of all four algorithms on 'all features' and 'triaxial Staudenmayer features' extracted from raw acceleration signal (30 Hz). The 'all features' representation had a significant improvement over the 'triaxial Staudenmayer features' (one way ANOVA, p-value < 0.05). The other models were shown to have significant improvements over ANN on 'all features' (t-test, p-value < 0.05).

Logistic regression with L1 regularization (Glmnet+L1) achieved the highest mean classification accuracy given either feature set. It achieved a mean accuracy of 0.8906 on 'all features', which was also the highest mean classification accuracy reported in our experiments, and a mean classification accuracy of 0.8519 on 'triaxial Staudenmayer features'. The ANN was the second best model with 'triaxial Staudenmayer features' (0.8519), but with 'all features', its accuracy dropped to 0.3731. Both logistic regression with L2 regularization (Glmnet+L2) and adaboost with decision trees (Adaboost+J48) performed better than ANN on 'all features', but they could not outperform Glmnet+L1 on 'all features'.

2.2.5 Discussion

In this chapter, various data, feature and algorithm combinations were explored in order to improve the classification accuracy on physical activity from hip accelerometer data. The results suggested that the logistic regression with L1 regularization based on 'all features' extracted from raw accelerometer signal was the best performing setting. Compared to 'uniaxial Staudenmayer features', 'triaxial Staudenmayer features' resulted in higher classification accuracies for ANN models with 1 Hz data. This agreed with De Vries's conclusion that the models developed with triaxial features performed better than their uniaxial counterparts [10]. However, uniaxial and triaxial features did not result in a significant difference in performance for ANN models based on 30 Hz data. In our experiments, an accuracy of 0.8519 was the best ANN models could achieve. Further experiments even showed that a feature set containing more information, and inevitably more noise, even significantly deteriorated the performance of ANN models.

Raw accelerometer signal (30 Hz) seemed to always result in higher classification accuracies than processed data (1 Hz), likely due to the fact that some distinct information in the raw data might be lost as the data were processed into counts per second. It was quite surprising that the raw data could achieve much higher classification accuracies since most researchers developed their models based on processed data instead of raw data.

The results of our experiments suggested that, in general, increasing number of features and data resolution always had positive effects on classification accuracy. However, the comparison between algorithms under the best case scenario (all features, 30 Hz),

showed that the performance also depended on the regularization capabilities of models. Not every model could benefit from so many features, many of which were correlated. The ANN models performed well on triaxial Staudenmayer features, but with 'all features', its performance dropped rapidly. The features fed to ANN models were highly correlated and for certain activities, some features were irrelevant. The models could get misled with these features, and performed much worse than the same models based on less features. On the other hand, the logistic regression models used L1 regularization (lasso penalty) as a way of preventing overfitting, and the decision trees used as the base classifiers in adaboost also use pruning to prevent overfitting. L1 regularization worked better than L2 regularization as Glmnet+L1 outperformed Glmnet+L2 in every test. Adaboost+J48 demonstrated a relatively large variance in performance.

In conclusion, for our accelerometer-based physical activity data, a higher sampling frequency and a larger feature set did improve the performance of classifiers. However, in order to benefit from such large amount of information, the classifiers must be able to prevent overfitting through a strong form of regularization. In our case, the L1 regularization helped logistic regression take advantage of the higher sampling frequency and the larger feature set, and achieved the highest classification accuracy. Lacking such capability made ANN the weakest classifier under the same setting.

Chapter 3: Predicting Physical Activity with a Subwindow Ensemble Model

3.1 Introduction

Time series data can be classified by dividing the sequence into shorter windows (eg. 10 second windows), generating a feature vector representation for each window, and classifying these feature vectors as if they were data instances. A standard feature representation for these windows is to use summary statistics computed from the entire window (eg. the mean, the 25^{th}, 50^{th}, and 75^{th} percentiles). However, these summary features may be at too coarse a resolution and may not capture the discriminative aspects of different physical activities. Rather than committing to the granularity of the entire window, we propose generating different sets of features from smaller subwindows of the original window. We then train an ensemble of classifiers using each set of features and combine their predictions in a meta model.

3.2 Algorithm

Figure 3.1: Decomposing a time series of a 10-second walk into 1, 5 and 10-second overlapping subwindows. The subwindows shift by 1 second.

We call our approach the Subwindow Ensemble Model (SWEM), which was inspired by the spatiotemporal exploratory model (STEM) [13]. Figure 3.2 gives an overview of the structure of the SWEM. To train a SWEM, the time series are first decomposed

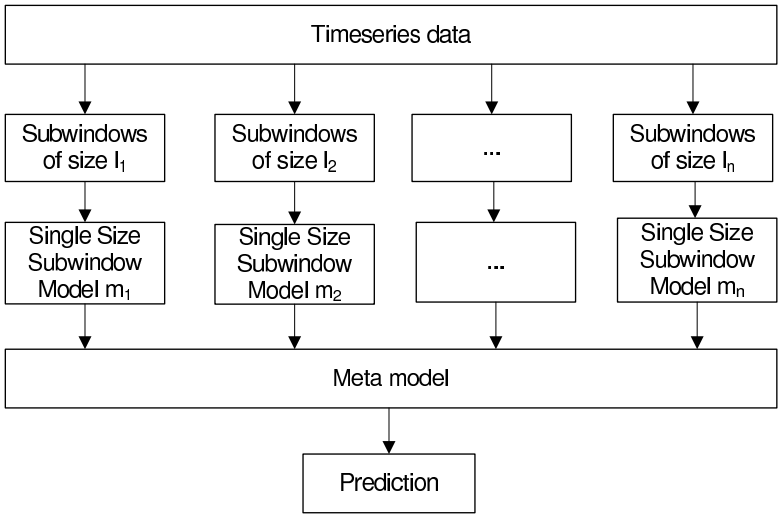

Figure 3.2: An overview of the structure of the Subwindow Ensemble Model (SWEM)

into overlapping subwindows. Figure 3.1 shows an example of decomposing a 10-second time series into overlapping subwindows of 3 different sizes. Then, for each size, a classic feature-based time series model is developed based on the corresponding subwindows. Note that the labels of the subwindows are the same as the label of the original time series. The predictions by the ensemble members are later combined by a meta model to make the final prediction.

The SWEM is designed for a specific type of time series which are composed of repetitive patterns. The example time series shown in Figure 3.1 is a time series of 10-second walk from the OSU_Hip dataset, which collects the accelerometer measurement of human physical activities (sitting, walking, running, etc.). The example time series is clearly composed of repetitive patterns each corresponding to a walk cycle. Intuitively, it makes sense to label all the subwindows as 'walk' as long as each component contains at least a complete walk cycle.

Algorithm 1 demonstrates how the ensemble members are developed. Function True-Label(t) returns the true label of a time series t. Function Featurize(t) generates the feature-based representation of subwindow t. Function Build-Model($training\_data$) trains a model based on $training\_data$.

Algorithm 2 shows the training process of the meta model. The features for the meta

Algorithm 1 MEMBER-TRAIN(T, L)

Input

T: time series for training.
L: subwindow sizes.

Output

M: the ensemble members.

Procedure

```
M ← {}
for each l in L do
    training_data ← {}
    for each time series t in T do
        label ← TRUE-LABEL(t)
        for each subwindow s of t with length l do
            x ← FEATURIZE(s)
            training_data ← training_data ∪ (x, label)
        end for
    end for
    M ← M ∪ BUILD-MODEL(training_data)
end for
return M
```

model are predictions by ensemble members. More specifically, each predictor variable is the majority vote of all subwindows of a certain size predicted by the corresponding ensemble member. In our pseudocode, p stores the predictions of subwindows of a certain size by an ensemble member. Function $MajorityVote(p)$ returns the majority prediction of the subwindows. v stores the predictions by all ensemble members, and function $Feature-Vector(v)$ converts v to a feature vector used to train the meta model.

Algorithm 3 shows the prediction procedure of SWEM. To predict a time series, the SWEM first decomposes it to subwindows, and then predicts the subwindows with the ensemble members. Then, the meta model predicts the time series based on the predictions of the ensemble members.

Algorithm 2 META-TRAIN(M, T, c)

Input

1: M: the ensemble members generated by MEMBER-TRAIN.
2: T: time series for training.
3: c: the number of classes.

Output

1: *meta*: the meta model.

Procedure

```
training_data ← {}
for each time series t in T do
  label ← TRUE-LABEL(t), v ← {}
  for each ensemble member m from M do
    l ← LENGTH(m), p ← {}
    for each subwindow s of t with length l do
      p ← p ∪ PREDICT(m, FEATURIZE(s))
    end for
    v[m] ← MAJORITYVOTE(p)
  end for
  x ← FEATURE-VECTOR(v)
  training_data ← training_data ∪ (x, label)
end for
meta ← BUILD-MODEL(training_data)
return meta
```

3.3 Evaluation

The SWEM was evaluated on six time series datasets – three accelerometer-based physical activity datasets and three datasets from other domains. In addition, the SWEM was evaluated against the 1-nearest neighbor algorithm, which is a widely used benchmark in time series classification. As a second baseline, the performance of the SWEM was compared to that of its ensemble members individually, which we will refer to as the Single Size Subwindow Models (SSSMs). These SSSMs use a feature representation derived from a single subwindow size. Note that in our experiment, the largest subwindow was the entire time series, so the SSSM of the largest subwindow size was actually the classic feature-based approach we evaluated in previous chapter.

Algorithm 3 PREDICT($M, meta, T, C$)

Input

1: M: the ensemble members generated by MEMBER-TRAIN.
2: $meta$: the meta model trained with the prediction results of M.
3: t: the time series to be predicted.
4: c: the number of classes.

Output

1: *prediction*: the prediction of time series t.

Procedure

```
v ← {}
for each ensemble member m in M do
  l ← LENGTH(m), p ← {}
  for each subwindow s of t with length l do
    p ← p ∪ PREDICT(m, FEATURIZE(s))
  end for
  v[m] ← MAJORITYVOTE(p)
end for
x ← FEATURE-VECTOR(v)
prediction ← PREDICT(meta, x)
return prediction
```

3.3.1 Datasets

Six time series datasets were chosen to test our models. Three datasets, OSU_Hip, OSU_Wrist and HASC contained time series of physical activities measured by accelerometers mounted on subjects' body. UCR_ECG200, UCR_CricketX and UCR_Sony were chosen from the UCR time series datasets [24].

3.3.1.1 OSU Physical Activity Data (OSU_Hip & OSU_Wrist)

The subjects were asked to perform twelve trials in a controlled lab environment. The twelve trials were categorized into seven activity classes: lying down, sitting, standing & household chores, walking, running, basketball and dance. Details of the activity classes can be found in Table 3.1. The data were recorded by triaxial accelerometers mounted on the subjects' hips and wrists at a 30 Hz sampling rate. Predictions were based on 10-second time windows (i.e. each time series instance was 10 seconds in duration).

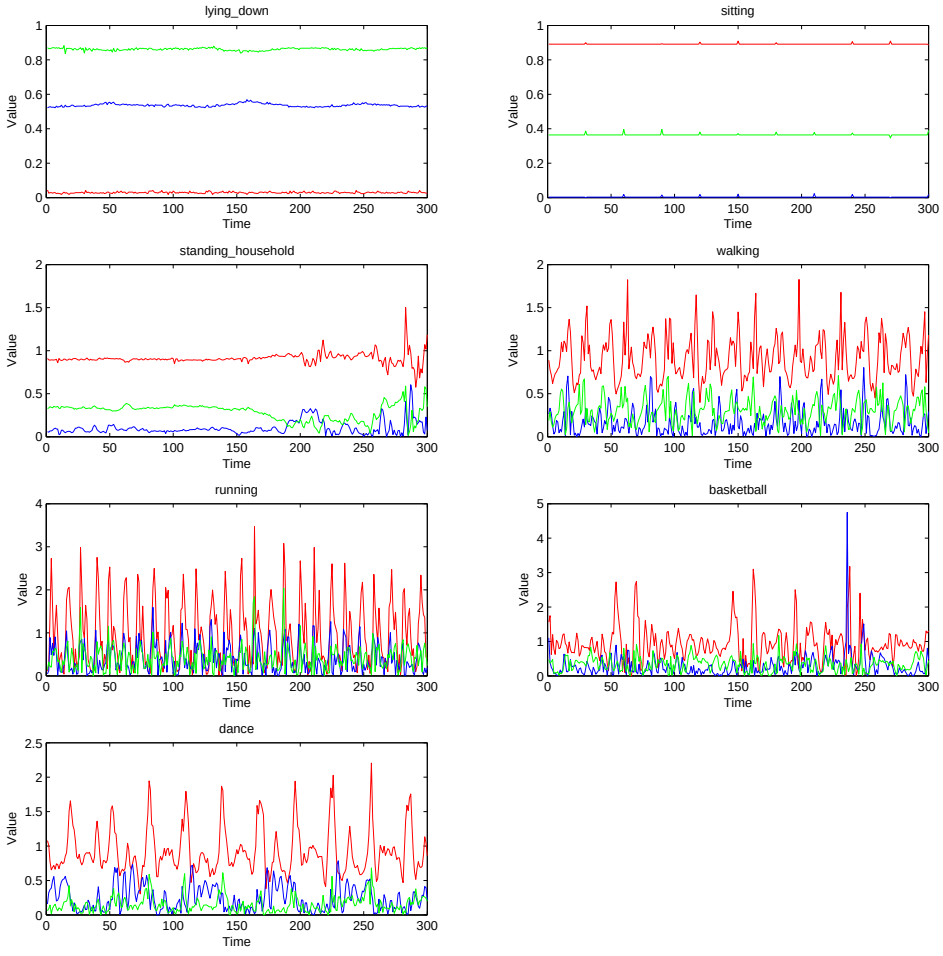

Figure 3.3: An example of all seven classes in the OSU_Hip dataset. These plots illustrate triaxial accelerometer data collected at 30 Hz, and each color in a plot represents one axis. Each time series instance lasts for 10 seconds.

Activity Class	Trial	Activity	Description of Activity
lying down	1	Supine Resting	Lie on floor mat or cot in supine position. Awake with arms at side. Instructed to minimize all bodily movements.
sitting	2	Hand writing	While sitting in a chair at a desk, use a ball point pen and a pad of paper to transcribe a standardized script.
	7	Video game	Sit in chair and play video game.
standing & household chores	4	Throw and Catch	Throw and catch a ball while standing 5-10 ft from a research assistant. 15 throws per min.
	3	Laundry Task	Load a laundry basket with towels and carry it 10 feet; then they dump out the towels, fold them, load them back in the basket, and carry it back to the original starting spot.
	8	Sweeping Floor	Sweep confetti on floor continuously using broom to a specified location and repeating.
walking	5	Comfortable walk	Walk at a self-selected comfortable speed around the perimeter of a gymnasium (1 lap = 63 m) .
	9	Brisk walk 1	Walk at a self-selected brisk speed around the perimeter of a gymnasium (1 lap = 63m) .
	12	Brisk walk 2	Walk on a treadmill at speed equal to that achieved during the brisk over-ground walking trial.
running	11	Run	Run at a self-selected speed around the perimeter of a gymnasium (1 lap = 63m) .
basketball	10	Basketball	Shoot a basketball using an 8 ft or regulation hoop. Instructed to shoot the ball, get the rebound and chase after the ball continuously.
dance	6	Aerobics	Follow a simple aerobics video. Routine included simple arm and leg movements.

Table 3.1: Activity class descriptions in OSU physical activity data

The absolute value of the accelerometer readings was used. OSU_Hip refers to the data collected from hips, and OSU_Wrist refers to the data collected from wrists. Figure 3.3 shows an example of all seven classes in the OSU_Hip dataset. The time series in the OSU_Wrist dataset are very similar to the time series from the OSU_Hip dataset.

The example time series of 'lying down', 'sitting', 'walking', 'running' and 'dance' from the OSU_Hip dataset clearly consisted of repetitive patterns, while the 'standing & household chores' and 'basketball' time series were less homogeneous. Note that the time series were three dimensional since they were recorded by triaxial accelerometers. The three axes appeared to be highly correlated. The SWEM was expected to perform well on these two datasets.

3.3.1.2 HASC

The HASC (Physical Activity Sensing Consortium) aimed at constructing a large scale database for studying physical activity through sensing [20, 32, 21]. The dataset used in this experiment can be found at the official website of HASC[1]. The Sample Data was used in our experiments. The data were collected from seven subjects with triaxial accelerometers at a 100 Hz sampling rate. Six activities: 'stay', 'walk', 'jog', 'skip', 'stUp' (stair-up) and 'stDown' (stair-down) were performed by all subjects in a controlled lab environment. The predictions were based on 10-second time windows as well. Figure 3.4 shows an example of all six classes in the HASC dataset. The repetitive patterns are obvious in all six activities. The SWEM was expected to perform well on this dataset as well.

3.3.1.3 UCR Time Series Datasets

The UCR Time Series Datasets [24] provides an extensive collection of time series datasets from various fields to test classification/clustering algorithms for time series data. Three datasets were selected from the UCR Time Series Datasets to test the performance of our Subwindow Ensemble Model.

[1] http://hasc.jp/hc2011/download-en.html

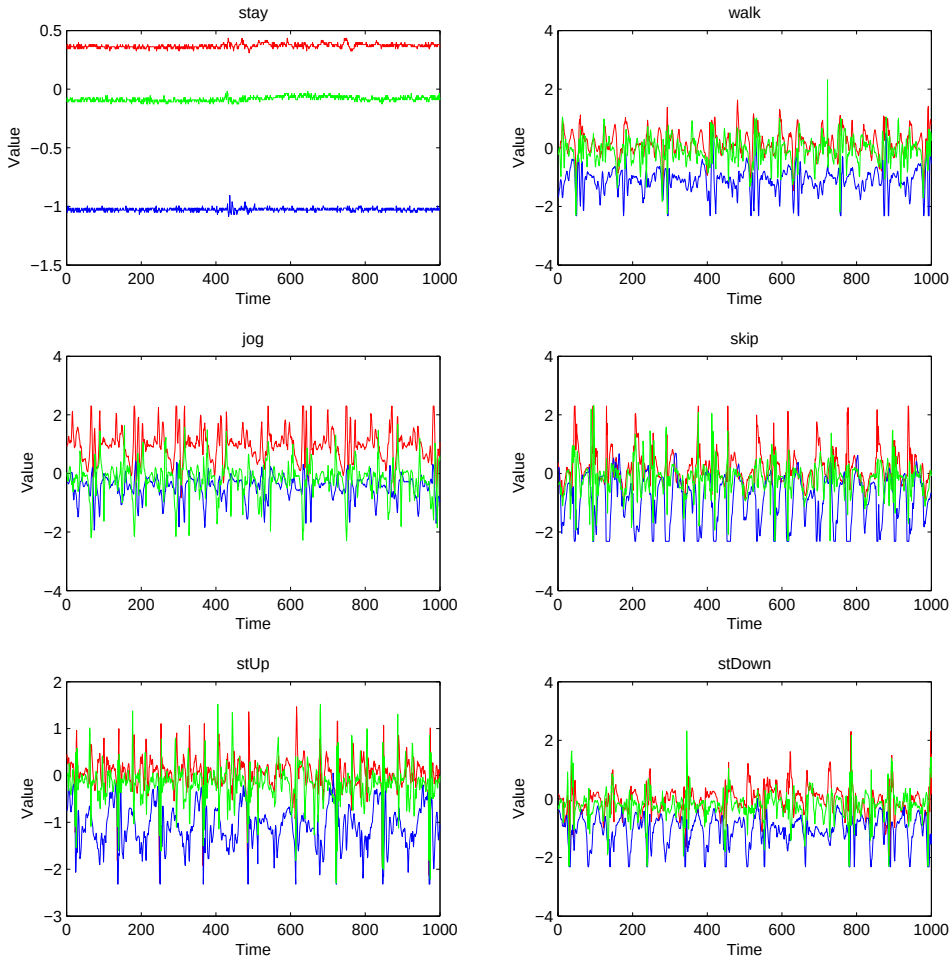

Figure 3.4: An example of all six classes in the HASC dataset. These plots illustrate triaxial accelerometer data collected at 100 Hz, and each color in a plot represents one axis. Each time series instance lasts for 10 seconds.

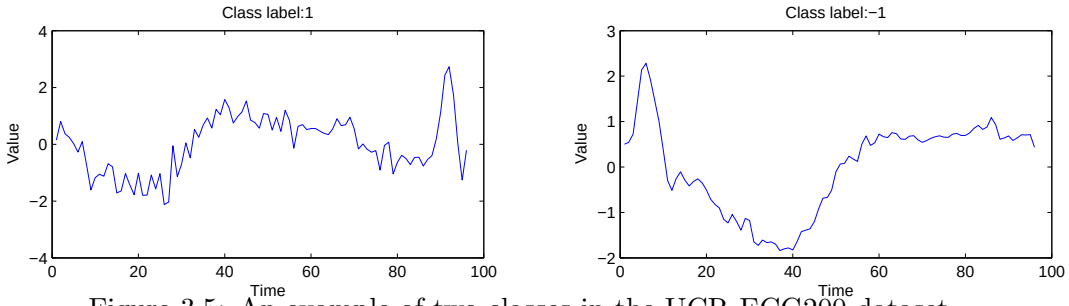

Figure 3.5: An example of two classes in the UCR_ECG200 dataset.

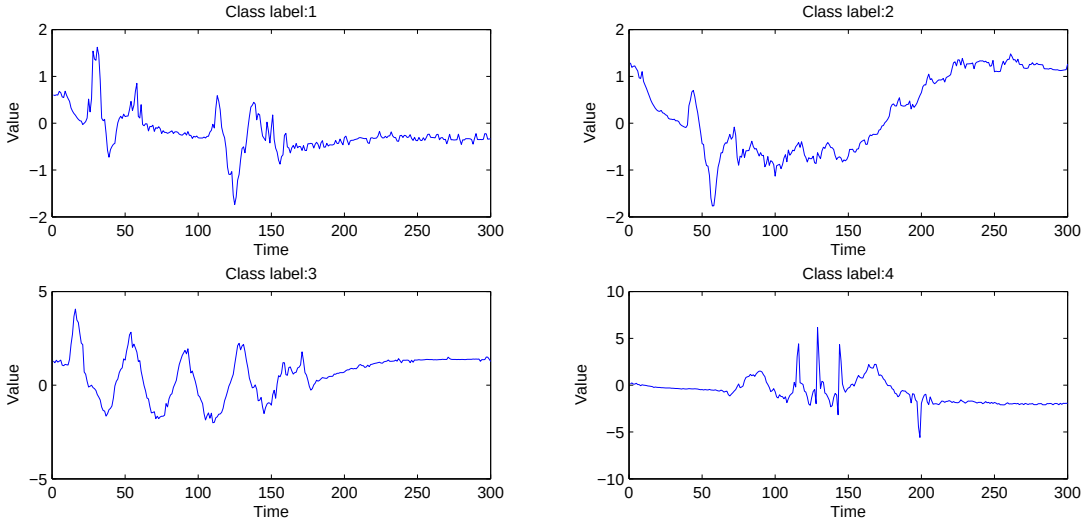

Figure 3.6: An example of four (out of twelve) classes in the UCR_CricketX dataset.

1. **UCR_ECG200 (ECG200):** Each time series in the dataset was the measurement of cardiac electrical activity recorded by one electrode during one heartbeat, and a label of normal or abnormal was assigned to each time series instance [33]. Figure 3.5 shows an example of the two classes in the UCR_ECG200 dataset.

2. **UCR_CricketX (Cricket_X):** Time series in the UCR_CricketX were accelerometer signals of the cricket (a very popular game in British Commonwealth countries) umpires performing twelve different signals used in the game of cricket, and the data were collected from accelerometers mounted on wrists [1]. The UCR_CricketX dataset only contained the accelerometer signals in x-axis. Figure 3.6 shows an example of the four of the twelve classes in the UCR_CricketX dataset.

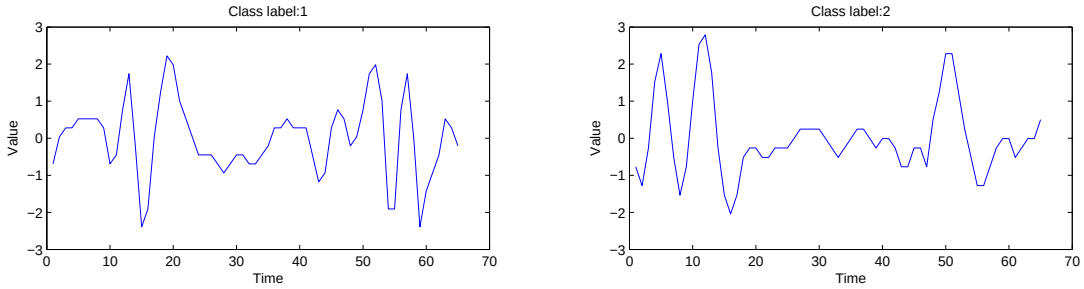

Figure 3.7: An example of two classes in the UCR_Sony dataset.

3. **UCR_Sony (SonyAIBORobot SurfaceII):** Time series in the UCR_Sony dataset were x-axis accelerometer signals of the SONY AIBO Robots (a small, dog-shaped, quadruped robot that comes equipped with multiple sensors) walking on two different surfaces: carpet and cement, and each time series represented one walk cycle[1]. The dataset was created by [47]. Figure 3.7 shows an example of the two classes in the UCR_Sony dataset.

The time series of the three datasets from the UCR time series datasets were very different from the time series from the OSU_Hip, the OSU_Wrist and the HASC datasets, and did not have the obvious repetitive patterns from these three datasets. Since the SWEM was not designed for this kind of time series, we did not expect it to perform well.

3.3.2 Experiments

3.3.2.1 Subwindow Ensemble Model

For the OSU_Hip, OSU_Wrist and HASC datasets, the predictions were made on 10-second time windows. 10 ensemble members were developed with 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10-second subwindows. For the UCR time series datasets, however, since the time units were unknown, the size of subwindows were measured by the number of time ticks. For the UCR_ECG200 dataset, ensemble members were developed with 10, 20, 30, 40, 50, 60, 70, 80 and 96-time-tick subwindows. For the UCR_CricketX dataset, ensemble members were developed with 30, 60, 90, 120, 150, 180, 210, 240, 270 and 300-time-tick subwindows. Finally, for the UCR_Sony dataset, the ensemble members were developed

with 10, 20, 30, 40, 50 and 65-time-tick subwindows.

For the SWEM and its feature-based baseline models, we selected the same features described in Table 2.1 as these features had been proven to work well for algorithms with regularization in previous chapter. Note that these features were simple, and could be computed efficiently (in linear time or even less). For the OSU_Hip, the OSU_Wrist and the HASC datasets, which were triaxial accelerometer data, Feature 1-14 were extracted from each axis, and Feature 15 (the correlation between axes) was extracted from each pair of axes. For other datasets, only Feature 1-14 were extracted.

The logistic regression models with L1 regularization (lasso penalty) and support vector machines (linear kernel) were used in our experiments as both the ensemble members and the meta model because algorithms with regularization to prevent from overfitting had been proven to be able to maximize the benefits from a large and comprehensive time series feature set. The 'glmnet'[15] package was used as the logistic regression implementation. The 'e1071'[12] package provides an interface in R[36] for libsvm[8], and was used as the SVM implementation in our experiments. The logistic regression based SWEM is referred to as SWEM_GL1, and the SVM based SWEM is referred to as SWEM_SVM.

The dataset was randomly split into three non-overlapping subsets for training, validation and testing. The models were trained on the training data, and the ones achieving the highest classification accuracy on the validation data were chosen as the best tuned model to be tested on the testing data. The testing results were reported as the final evaluation of that model. For the logistic regression models used in SWEM, the λ parameter (the penalty coefficient) was tuned. The range of λ was automatically generated by 'glmnet' during training. L1 regularization (lasso penalty) was used. For the SVM used in SWEM, the cost parameter (0.01, 0.1, 1, 10, 100 and 1000) was tuned. Each model was evaluated using 30 training-validation-testing splits as described above. The average accuracy of 30 iterations was reported as the final evaluation for that model.

3.3.2.2 1-Nearest neighbor Algorithm

The 1-nearest neighbor algorithm is widely used as a baseline to compare against new algorithms for time series classification. In our experiment, the SWEM were compared to the 1-nearest neighbor algorithm on the OSU, HASC and UCR datasets. The original

Euclidean distance and DTW (dynamic time warping) were used to compare the distance between two time series instances. The 'dtw'[44, 16] package in R was used in this experiment to calculate DTW.

Let time series $A = \{a_1, a_1, \ldots, a_n\}$, time series $B = \{b_1, b_2, \ldots, b_n\}$, $D(A, B)$, the Euclidean distance between A and B is

$$D(A, B) = \sqrt{\sum_{i=1}^{n} d(a_i, b_i)^2}$$

where $d(a_i, b_i)$ is the euclidean distance between a_i and b_i. For one-dimensional time series (UCR), $d(a_i, b_i) = |a_i - b_i|$. For three-dimensional time series (OSU_Hip, OSU_Wrist, and HASC), a_i and b_i are three-dimensional vectors: $a_i = \{a_{ix}, a_{iy}, a_{iz}\}$, $b_i = \{b_{ix}, b_{iy}, b_{iz}\}$, and $d(a_i, b_i)$ is defined as

$$d(a_i, b_i) = \sqrt{(a_{ix} - b_{ix})^2 + (a_{iy} - b_{iy})^2 + (a_{iz} - b_{iz})^2}$$

The DTW algorithm searches the optimal alignment between two time series, and the distance after warping is

$$D(A, B) = \sqrt{\sum_{i=1}^{n} d(a_{I_a[i]}, b_{I_b[i]})^2}$$

where $\{I_a[i], I_b[i]\}$ is the alignment (the $I_a[i]^{th}$ data point in time series A matches the $I_b[i]^{th}$ data point in time series B).

NN_EUC refers to the 1-nearest neighbor model based on the original Euclidean distance, and NN_DTW refers to the 1-nearest neighbor model based on DTW.

In order to make the experiment results of the 1-nearest neighbor models comparable to the results of the feature-based models, the data used to train the 1-nearest neighbor models were exactly the same as the data used to train the feature-based models, and the testing data were exactly the same as well. Since no parameters need to be tuned for the 1-nearest neighbor models, the validation step was skipped. The average accuracy of 30 iterations was reported as the final evaluation for 1-nearest neighbor models.

3.3.3 Results

3.3.3.1 The Subwindow Ensemble Model vs the 1-Nearest Neighbor Models

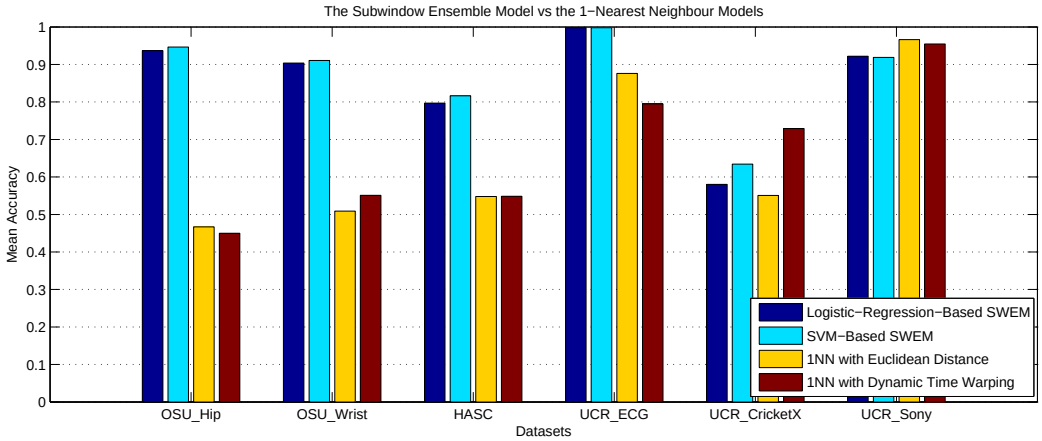

Figure 3.8: Mean Classification Accuracies of the Subwindow Ensemble Model and the 1-Nearest Neighbor Models

Table 3.2 and Figure 3.8 show the results of the Subwindow Ensemble Models vs the 1-nearest neighbor models. The SVM based SWEM (SWEM_SVM) was the best performing model on the OSU_Hip (0.9465), OSU_Wrist (0.9106) and HASC (0.8165) datasets. The logistic regression based SWEM (SWEM_GL1) was the second best model on these three datasets. SWEM_GL1 achieved the highest classification accuracy on the UCR_ECG200 dataset (0.9979). However, on the other two UCR datasets, the 1-nearest neighbor models outperformed the SWEMs.

3.3.3.2 The Subwindow Ensemble Model vs the Single Size Subwindow Models

Table 3.3 and Figure 3.9 show the classification accuracies of the SVM based Subwindow Ensemble Model and the Single Size Subwindow Models on the OSU_Hip dataset. Table 3.4 and Figure 3.10 show the classification accuracies of these models on the OSU_Wrist dataset. Figure 3.5 and Figure 3.11 show the classification accuracies of these models on the HASC dataset. SUB1_SVM indicates the SVM based Single Size

Dataset	Model	Mean of Accuracy	SD of Accuracy
OSU_Hip	SWEM_GL1	0.9367	0.0164
	SWEM_SVM	**0.9465**	0.0109
	NN_EUC	0.4669	0.0308
	NN_DTW	0.4499	0.0354
OSU_Wrist	SWEM_GL1	0.9038	0.0244
	SWEM_SVM	**0.9106**	0.0198
	NN_EUC	0.5089	0.0864
	NN_DTW	0.5510	0.0602
HASC	SWEM_GL1	0.7970	0.0301
	SWEM_SVM	**0.8165**	0.0291
	NN_EUC	0.5480	0.0304
	NN_DTW	0.5485	0.0298
UCR_ECG200	SWEM_GL1	**0.9979**	0.0068
	SWEM_SVM	**0.9979**	0.0068
	NN_EUC	0.8760	0.0414
	NN_DTW	0.7948	0.0357
UCR_CricketX	SWEM_GL1	0.5803	0.0240
	SWEM_SVM	0.6343	0.0225
	NN_EUC	0.5507	0.0221
	NN_DTW	**0.7291**	0.0253
UCR_Sony	SWEM_GL1	0.9220	0.0134
	SWEM_SVM	0.9187	0.0119
	NN_EUC	**0.9663**	0.0096
	NN_DTW	0.9547	0.0148

Table 3.2: Average classification accuracies of the Subwindow Ensemble Model and the 1-Nearest Neighbor Models. The bold font marks the model with the highest classification accuracy on the corresponding dataset.

Model	Accuracy	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.9465	0.9806	0.9423	0.9678	0.9541	0.9823	0.9419	0.8041
SUB1_SVM	0.9219	0.9709	0.9294	**0.9893**	0.9488	**0.9876**	0.7398	0.6931
SUB2_SVM	0.9402	0.9735	0.9271	0.9836	0.9543	0.9844	0.8931	0.7648
SUB3_SVM	0.9419	0.9719	0.9365	0.9727	0.9502	0.9870	0.9283	0.7756
SUB4_SVM	0.9408	0.9800	0.9265	0.9709	0.9533	0.9810	0.9178	0.7861
SUB5_SVM	0.9401	0.9780	0.9357	0.9564	0.9494	0.9811	**0.9407**	0.7931
SUB6_SVM	0.9411	0.9787	0.9299	0.9609	0.9572	0.9798	0.9306	0.7911
SUB7_SVM	0.9422	0.9802	0.9353	0.9519	0.9565	0.9798	0.9378	**0.8131**
SUB8_SVM	0.9420	**0.9819**	0.9296	0.9608	**0.9615**	0.9776	0.9206	0.7991
SUB9_SVM	**0.9436**	0.9817	**0.9374**	0.9572	0.9567	0.9789	0.9359	0.8104
SUB10_SVM	0.9426	0.9772	0.9318	0.9666	0.9599	0.9776	0.9161	0.7978

Table 3.3: Activity classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the OSU_Hip dataset. The bold font marks the highest classification accuracy of the SSSM for each activity as well as the highest overall accuracy of the SSSM.

Subwindow Model trained with features extracted 1-second subwindows, and SUB2_SVM indicates the SVM based Single Size Subwindow Model trained with features extracted 2-second subwindows, and so on. The mean classification accuracies of these models on individual activities are also shown.

Our results show that the Subwindow Ensemble Model consistently performed better than any of the Single Size Subwindow Models. In addition, the Single Size Subwindow Models achieved very different performance on different datasets. For example, on the OSU_Hip dataset, SUB9_SVM was the best SVM based SSSM, and achieved a mean classification accuracy of 0.9436. Whereas on the OSU_Wrist dataset, SUB5_SVM was the best one, and achieved a mean classification accuracy of 0.9080. On the HASC dataset, the best SVM based Single Size Subwindow Model was SUB2_SVM, which achieved a mean classification accuracy of 0.8150.

3.3.4 Discussion

The results show that the SWEM worked very well on the accelerometer-based physical activity datasets (OSU_Hip, OSU_Wrist and HASC). The success of the SWEM on these three dataset was expected as it was designed for the time series composed of repetitive

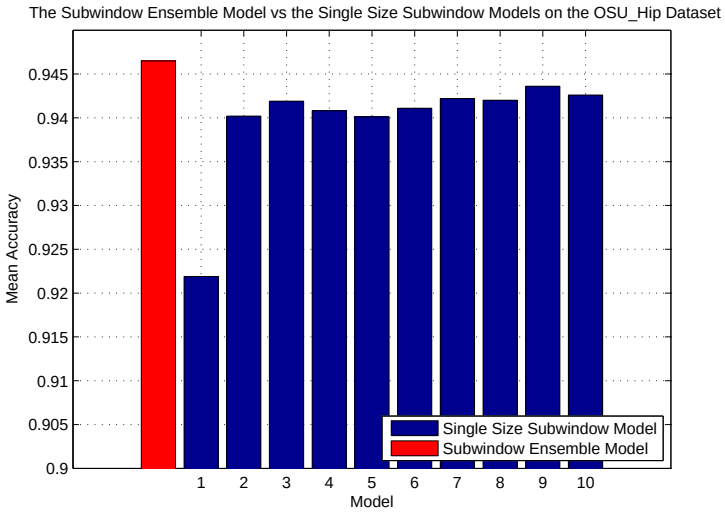

Figure 3.9: Classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the OSU_Hip dataset.

patterns under the hypothesis that the most discriminative features could be found in subwindows of various sizes. The high classification accuracies achieved by the SWEMs on the UCR_ECG200 dataset were probably due to one very distinctive feature, the coefficients of variation: $c_v = \frac{\sigma_s}{\mu_s}$, extracted from the largest subwindow (the entire time series). In this particular dataset, this feature was much more informative than the distance measures we tested.

The best performing model on the UCR_CricketX dataset was NN_DTW (the 1-nearest neighbor model with dynamic time warping), and the best performing model on the UCR_Sony dataset was NN_EUC (the 1-nearest neighbor model with the Euclidean distance). They both performed significantly better than the SWEM. The UCR_Sony and the UCR_CricketX datasets were characterized by their overall shape, but the important temporal characteristics of these time series were ignored by the SWEM. Another problem is that in these cases, the subwindows cannot represent the entire time series. For example, when decomposing a time series from UCR_CricketX into multiple subwindows of a certain size, none of the subwindows would be sufficient to represent the complete signal performed by the umpire. One possible solution for this issue is to preserve the information about the order of the subwindows in the original time series when the ensemble members are trained, so the temporal characteristics of the time series could be used by the classifier.

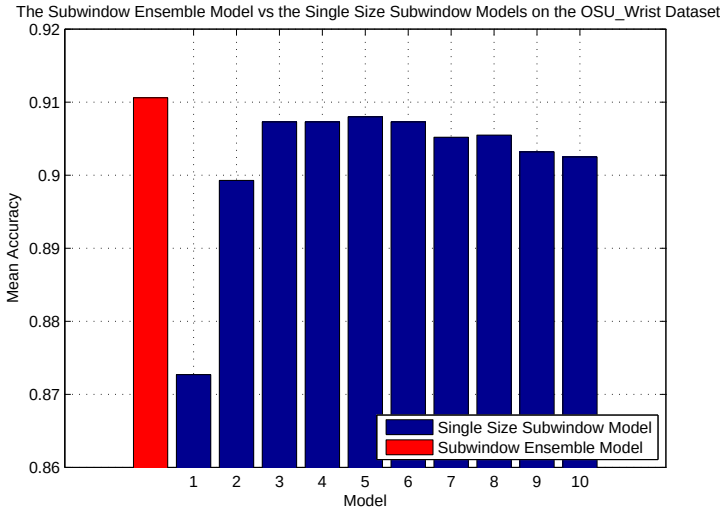

Figure 3.10: Classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the OSU_Wrist dataset.

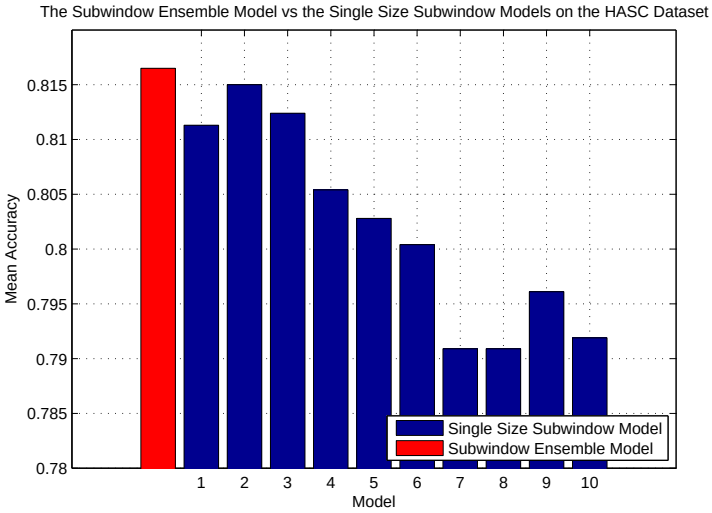

Figure 3.11: Classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the HASC dataset.

Model	Accuracy	Classification Accuracy of Each Physical Activity						
		lying	sitting	standing	walking	running	basketball	dance
SWEM_SVM	0.9106	0.7993	0.9522	0.9389	0.9562	0.8345	0.9072	0.7722
SUB1_SVM	0.8727	0.7368	**0.9608**	**0.9767**	0.9639	**0.8565**	0.4300	0.7257
SUB2_SVM	0.8993	0.7708	0.9571	0.9618	**0.9700**	0.8547	0.7206	0.7243
SUB3_SVM	0.9073	0.7924	0.9481	0.9464	0.9583	0.8507	0.8461	0.7694
SUB4_SVM	0.9073	0.7778	0.9559	0.9455	0.9563	0.8426	0.8828	0.7403
SUB5_SVM	**0.9080**	**0.8021**	0.9438	0.9368	0.9514	0.8281	**0.9194**	**0.7701**
SUB6_SVM	0.9073	0.7875	0.9549	0.9382	0.9534	0.8299	0.9056	0.7535
SUB7_SVM	0.9052	0.8007	0.9549	0.9239	0.9522	0.8218	0.9150	0.7611
SUB8_SVM	0.9055	0.7764	0.9605	0.9378	0.9567	0.8235	0.8933	0.7410
SUB9_SVM	0.9032	0.7903	**0.9608**	0.9192	0.9518	0.8241	0.9050	0.7576
SUB10_SVM	0.9025	0.7819	0.9590	0.9340	0.9575	0.8148	0.8822	0.7312

Table 3.4: Activity classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the OSU_Wrist dataset. The bold font marks the highest classification accuracy of the SSSM for each activity as well as the highest overall accuracy of the SSSM.

Although the SWEM failed on some time series datasets, it always had the advantage on computation time over the 1-nearest neighbor models. All features require only linear time to calculate. It is much faster than the 1-nearest neighbor algorithms to predict a new time series. This provides the SWEM with a huge advantage in real-world application where quick predictions are required.

The experiments on the SWEM and the SSSM also had very interesting results. The classification accuracy distributions of the SVM based SSSMs were very different on the OSU_Hip, OSU_Wrist and HASC datasets. On the OSU_Hip dataset, subwindows of various sizes (except 1-second subwindows) resulted in similar performance. On the OSU_Wrist dataset, the middle-sized subwindows led to higher classification accuracies than small or large size subwindows. On the HASC dataset, the overall trend was that as the size of subwindows increased, the classification accuracy decreased. The completely different behaviors of the SSSMs provided strong empirical evidence supporting our hypothesis that the most discriminative features could be extracted from subwindows of different sizes. The fact that the SWEM always outperformed the SSSMs suggests that combining the information extracted from subwindows of various sizes might be a good approach to improve classification accuracy.

Model	Accuracy	Classification Accuracy of Each Physical Activity					
		stay	walk	jog	skip	stUp	stDown
SWEM_SVM	0.8165	0.9956	0.7656	0.7989	0.8400	0.7111	0.7878
SUB1_SVM	0.8113	**1.0000**	0.7456	**0.8122**	0.8267	0.6800	0.8033
SUB2_SVM	**0.8150**	0.9989	0.7367	0.7956	**0.8456**	0.7067	**0.8067**
SUB3_SVM	0.8124	**1.0000**	0.7389	0.8044	0.8178	**0.7211**	0.7922
SUB4_SVM	0.8054	0.9989	0.7378	0.7911	0.8267	0.6978	0.7800
SUB5_SVM	0.8028	0.9944	0.7489	0.8000	0.8244	0.7000	0.7489
SUB6_SVM	0.8004	0.9933	0.7322	0.7889	0.8322	0.7000	0.7556
SUB7_SVM	0.7909	0.9944	0.7389	0.8033	0.8156	0.6567	0.7367
SUB8_SVM	0.7909	0.9867	0.7233	0.7878	0.8344	0.6689	0.7444
SUB9_SVM	0.7961	0.9878	**0.7767**	0.8078	0.8111	0.6767	0.7167
SUB10_SVM	0.7919	0.9811	0.7311	0.7944	0.8256	0.6689	0.7500

Table 3.5: Activity classification accuracies of the Subwindow Ensemble Model and the Single Size Subwindow Models (SVM based) on the HASC dataset. The bold font marks the highest classification accuracy of the SSSM for each activity as well as the highest overall accuracy of the SSSM.

In conclusion, the experiment results showed that using features from subwindows of various sizes was better than just using features from subwindows of a single size. The SWEM was able to figure out a linear combination of SSSM predictions to achieve a classification accuracy higher than any SSSM could achieve.

Chapter 4: Conclusion

The results from this thesis showed that machine learning algorithms that classify physical activity from accelerometer data achieved much higher accuracy when raw accelerometer signals from three axes were used and when the data was represented by a more comprehensive set of features. However, in order to benefit from this additional information, these algorithms needed to regularize their models heavily to prevent overfitting.

In the second part of this thesis, the SWEM was proposed to leverage the observation that the most discriminative features for physical activities exist at different resolutions. The results showed that the SWEM achieved the best performance on accelerometer-based physical activity data, but did not outperform common baseline algorithms on time series datasets that lacked repetitive patterns.

Bibliography

[1] A. Al Mueen. *Exact Primitives for Time Series Data Mining*. PhD thesis, University of California, Riverside, 2012.

[2] N. Armstrong and J.R. Welsman. The physical activity patterns of european youth with reference to methods of assessment. *Sports Medicine*, 36(12):1067–1086, 2006.

[3] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, pages 1–17, 2004.

[4] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*, volume 2, 1994.

[5] A.G. Bonomi, A.H.C. Goris, B. Yin, and K.R. Westerterp. Detection of type, duration, and intensity of physical activity using an accelerometer. *Medicine & Science in Sports & Exercise*, 41(9):1770, 2009.

[6] A.G. Bonomi, G. Plasqui, A.H.C. Goris, and K.R. Westerterp. Improving assessment of daily energy expenditure by identifying types of physical activity with a single accelerometer. *Journal of Applied Physiology*, 107(3):655–661, 2009.

[7] J. Carter, D. Wilkinson, S. Blacker, M. Rayson, J. Bilzon, R. Izard, A. Coward, A. Wright, A. Nevill, K. Rennie, et al. An investigation of a novel three-dimensional activity monitor to predict free-living energy expenditure. *Journal of sports sciences*, 26(6):553–561, 2008.

[8] C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[9] S.E. Crouter, K.G. Clowers, and D.R. Bassett Jr. A novel method for using accelerometer data to predict energy expenditure. *Journal of applied physiology*, 100(4):1324–1331, 2006.

[10] S.I. de Vries, M. Engels, and F.G. Garre. Identification of children's activity type with accelerometer-based neural networks. *Medicine & Science in Sports & Exercise*, 43(10):1994, 2011.

[11] S.I. de Vries, F.G. Garre, L.H. Engbers, V.H. Hildebrandt, and S. van Buuren. Evaluation of neural networks to identify types of activity using accelerometers. *Medicine & Science in Sports & Exercise*, 43(1):101, 2011.

[12] Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer, , and Andreas Weingessel. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*, 2011. R package version 1.6.

[13] D. Fink, W.M. Hochachka, B. Zuckerberg, D.W. Winkler, B. Shaby, M.A. Munson, G. Hooker, M. Riedewald, D. Sheldon, and S. Kelling. Spatiotemporal exploratory models for broad-scale survey data. *Ecological Applications*, 20(8):2131–2147, 2010.

[14] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.

[15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[16] Toni Giorgino. Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, 31(7):1–24, 2009.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[18] J. He, H. Li, and J. Tan. Real-time daily activity classification with wireless sensor networks using hidden markov model. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3192–3195. IEEE, 2007.

[19] D. John, S. Liu, JE Sasaki, CA Howe, J. Staudenmayer, RX Gao, and PS Freedson. Calibrating a novel multi-sensor physical activity measurement system. *Physiological Measurement*, 32:1473, 2011.

[20] N. Kawaguchi, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara, Y. Sumi, and N. Nishio. Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings. In *Proceedings of the 2nd Augmented Human International Conference*, page 27. ACM, 2011.

[21] N. Kawaguchi, H. Watanabe, T. Yang, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, H. Hada, S. Inoue, et al. Hasc2012corpus: Large scale human activity corpus and its application. 2012.

[22] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.

[23] E. Keogh and C.A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[24] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C.A. Ratanamahatana. The ucr time series classification/clustering homepage (2011) `http://www.cs.ucr.edu/~eamonn/time_series_data`.

[25] Z. Li. Exercises intensity estimation based on the physical activities healthcare system. In *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*, volume 3, pages 132–136. IEEE, 2009.

[26] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.

[27] S. Liu, R. Gao, and P. Freedson. Computational methods for estimating energy expenditure in human physical activities. *Medicine and science in sports and exercise*, 2012.

[28] S. Liu, R.X. Gao, and P. Freedson. Design of a wearable multi-sensor system for physical activity assessment. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 254–259. IEEE, 2010.

[29] S. Liu, R.X. Gao, D. John, J. Staudenmayer, and P.S. Freedson. Svm-based multi-sensor fusion for free-living physical activity assessment. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 3188–3191. IEEE, 2011.

[30] T. Matsumura, VT Chemmalil, ML Gray, JE Keating, RL Kieselbach, SB Latta, N. Occhialini, E. Kinnal, S. O'Toole, and RA Peura. Device for measuring real-time energy expenditure by heart rate and acceleration for diabetic patients. In *Bioengineering Conference, 2009 IEEE 35th Annual Northeast*, pages 1–2. IEEE, 2009.

[31] A. Mueen, E. Keogh, and N. Young. Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1154–1162. ACM, 2011.

[32] N. Ogawa, K. Kaji, and N. Kawaguchi. Effects of number of subjects on activity recognition-findings from hasc2010corpus. In *International Workshop on Frontiers in Activity Recognition using Pervasive Sensing (IWFAR2011)*, pages 48–51, 2011.

[33] R.T. Olszewski. Generalized feature extraction for structural pattern recognition in time-series data. Technical report, DTIC Document, 2001.

[34] D.M. Pober, J. Staudenmayer, C. Raphael, and P.S. FREEDSON. Development of novel techniques to classify physical activity mode using accelerometers. *Medicine & Science in Sports & Exercise*, 38(9):1626, 2006.

[35] J.R. Quinlan. *C4. 5: programs for machine learning*. Morgan kaufmann, 1993.

[36] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[37] M.T. Richardson, B.E. Ainsworth, D.R. JacobsJR, and A.S. Leon. Validation of the stanford 7-day recall to assess habitual physical activity. *Annals of epidemiology*, 11(2):145–153, 2001.

[38] M.P. Rothney, M. Neumann, A. Béziat, and K.Y. Chen. An artificial neural network model of energy expenditure using nonintegrated acceleration signals. *Journal of Applied Physiology*, 103(4):1419–1427, 2007.

[39] N. Ruch, M. Rumo, and U. Mäder. Recognition of activities in children by two uniaxial accelerometers in free-living conditions. *European journal of applied physiology*, 111(8):1917–1927, 2011.

[40] N. Sazonova, R.C. Browning, and E. Sazonov. Accurate prediction of energy expenditure using a shoe-based activity monitor. *Medicine & Science in Sports & Exercise*, 43(7):1312, 2011.

[41] K.H. Schimitz, M. Treuth, P. Hannan, R. McMurray, K.B. Ring, D. Catellier, and R. Pate. Predicting energy expenditure from accelerometry counts in adolescent girls. *Medicine and science in sports and exercise*, 37(1):155, 2005.

[42] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, and P. Freedson. An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer. *Journal of Applied Physiology*, 107(4):1300–1307, 2009.

[43] SW Su, L. Wang, BG Celler, E. Ambikairajah, and AV Savkin. Estimation of walking energy expenditure by using support vector regression. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3526–3529. IEEE, 2005.

[44] Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11–34, 2008.

[45] S.G. Trost, W.K. Wong, K.A. Pfeiffer, and Y. Zheng. Artificial neural networks to predict activity type and energy expenditure in youth. *Medicine and science in sports and exercise*, 2012.

[46] C.E. Tudor-Locke and A.M. Myers. Challenges and opportunities for measuring physical activity in sedentary adults. *Sports Medicine*, 31(2):91–100, 2001.

[47] D. Vail and M. Veloso. Learning from accelerometer data on a legged robot. In *Proceedings of the 5th IFAC/EURON symposium on intelligent autonomous vehicles*, 2004.

[48] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[49] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, pages 1–35, 2010.

[50] G. Welk. *Physical activity assessments for health-related research*, chapter Introduction to physical activity research, pages 3–18. Human Kinetics Publishers, 2002.

[51] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM, 2009.

[52] M. Zhang and A.A. Sawchuk. Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics*, pages 631–640. ACM, 2012.