

## Real-time Posture and Activity Recognition by SmartShoe

Christopher Hodapp<sup>1</sup>, S. Ryan Edgar<sup>2</sup>, George Fulk<sup>3</sup> and Edward Sazonov<sup>4+</sup>

<sup>1</sup> Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA

<sup>2</sup> Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY, USA

<sup>3</sup> Department of Physical Therapy, Clarkson University, Potsdam, NY, USA

<sup>2</sup> Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, USA

**Abstract.** Automatic recognition of physical activity and postural allocations such as standing or sitting can be used in behavioral modification programs aimed at minimizing static postures. We have developed and validated a footwear-based physical activity monitor (SmartShoe) that can reliably differentiate between most common postures and activities. In previous works, classification was performed using Support Vector Machines (SVM) which are computationally intensive and not well suited for the relatively low resources available on mobile devices such as cell phones. In this paper, we discuss a method for performing automatic posture classification using Artificial Neural Networks operating with fixed point precision arithmetic. The computational time is optimized through application of forward feature selection for determination of the most significant predictors. The method's performance is analyzed in terms of both throughput and accuracy and we compare it with the SVM classification. The results demonstrate feature selection picked the 61 most significant features out of 153, the proposed methodology reduces the computation time approximately 6000 times (from 1,347ms to 0.22ms) while maintaining comparable classification accuracy (95.2%).

**Keywords:** wearable sensors, physical activity monitoring, SmartShoe, obesity, stroke rehabilitation

### 1. Introduction

Obesity has reached epidemic proportions among American children, adolescents, and adults and continues to increase in prevalence [1]. Obesity puts sufferers at an increased risk of a number of diseases, including type-2 diabetes, cardiovascular disease, and even cancer. Obesity is usually caused by conditions where a person's energy expenditure is less than their energy intake over the long term. Strategies for combating obesity, then, tend to focus on lifestyle changes that engage subjects in an increased level of physical activity and thus increasing their energy expenditure. Specifically, it has been found that the posture assumed by subjects in their daily lives significantly affects energy expenditure [2].

Studies have shown that making patients aware of their daily physical activity levels results in an increased level of activity and a reduced BMI [3]. We seek to enable an approach to lifestyle change based on objective metrics and positive feedback. We believe widely-available mobile computing devices (smartphones) now have the potential to perform real-time recognition of postures and activities assumed by the subjects and provide behaviour-enhancing feedback. For this reason, we seek to create mobile applications and embedded hardware to enable self-monitoring of energy expenditure and postures each day. This may provide motivation and encouragement in choosing healthier lifestyle options.

Similar hardware and software platforms are also effective in monitoring of the activity levels of stroke patients during rehabilitation. Stroke, which occurs when the brain's blood supply is interrupted, affects nearly 800,000 people each year [4].

---

<sup>+</sup> Corresponding author. Tel.: +1-205-348-1981; fax: +1-205- 348-6959.  
E-mail address: esazonov@ieee.org.

In the past, we have focused on development of the SmartShoe sensor system and a PC-based platform for posture and activity classification methods [5]. SmartShoe was able to accurately recognize postures and activities in healthy [5] and post-stroke individuals [6], accurately quantify temporal gait parameters in the same two populations [7], and accurately predict caloric energy expenditure [8]. Now that SmartShoe has reached an acceptable level of maturity, the focus is on porting the classification methods on platforms with substantially lower power and computational resources.

The earliest developed method of posture and activity classification for SmartShoe was based on Support Vector Machines (SVMs) [5]. The SVM-based method allowed for very accurate posture recognition, but did not perform very well on the limited hardware offered by mobile phones. This paper explores feasibility of performing classification by Artificial Neural Networks (ANNs) operating with fixed point precision arithmetic. The computational time is further optimized through application of forward feature selection for determination of the most significant predictors. The method's performance is analyzed in terms of both throughput and accuracy in comparison with SVM classification.

## 2. Data

### 2.1. A Shoe-Based Wearable Sensor Array

The wearable sensor system (SmartShoe) was derived from the system which is described in detail in [6]. However, several changes have been made. First, rather than being embedded in the heel of the shoe, a wearable three-axis accelerometer (see Fig. 1) was clipped on the outsider side of the shoe. Also, the SmartShoe here used only three pressure sensors in its flexible insole (see Fig. 2), as opposed to the five in [5]. A pressure sensor was placed under the heel, the heads of the metatarsals, and the hallux.



Fig. 1: Sensor array mounted on shoe

### 2.2. Posture Classes

The ANN was trained to assign feature vectors to one of three posture classes (sitting, standing, and walking/jogging). These classes were selected due to their frequent occurrence in the daily lives of most humans.

### 2.3. Subjects

Our subjects were two healthy males between 20 and 40 years of age weighing between 87 and 110 kg.

### 2.4. Data Collection Method

The data were collected by SmartShoe system worn on a single (right) foot while subjects assumed each posture. The second foot was uninstrumented. Subjects were asked to sit and to stand for a time period of at least six minutes each. They were also asked to walk and to jog for a period of at least six minutes. They were also asked to make a conscious effort to perform different variations of each posture. For example, subjects were asked to sit with their feet both crossed and uncrossed, to shift their weight periodically while standing and to walk at different speeds and in different directions. Because twice as much time was spent in the “walk/jog” activity as was spent in the “sit” or the “stand” postures, the data from the latter were duplicated and incorporated in the final set twice. This was to prevent the training process from being biased toward classifying input vectors as belonging to the “walk/jog” class.



Fig. 2: Flexible insole with pressure sensors

## 3. Methods

### 3.1. Signal Processing

The time interval for feature vector computation was 0.2 seconds (features computed at 5 Hz). Sensor data were sampled at 100 Hz. Thus, each feature vector was based on 20 sensor readings. Five feature types were computed. First was the raw value of each sensor,  $\alpha_i$  (the values of the first accelerometer) through  $\omega_3$  (the values of the third pressure sensor). Second was the standard deviation of each sensor's value,  $\sigma(\alpha_i)$  (the standard deviation of all values for the first accelerometer) through  $\sigma(\omega_3)$  (the standard deviation of all values for the third pressure sensor). Third was the number of mean crossings in the value of each sensor,  $Z(\alpha_i)$  through  $Z(\omega_3)$ . Fourth was the sum of the values of the three pressure sensors in each sample,  $\Sigma(\omega_{*1})$  (the sum of all pressure sensor values in the first sample) through  $\Sigma(\omega_{*20})$  (the sum of the values of all pressure sensor values in the twentieth sample). Fifth was a Boolean feature which was true if the sum of the values of the pressure sensors exceeded a configurable threshold,  $\Sigma(\omega) > T$ , where T is based on the weight of the subject. In total, 153 features computed for each feature vector.

### 3.2. Classification By Artificial Neural Network

Fast Artificial Neural Network (FANN) library was used to add neural-network based classification functionality to the existing application codebase. FANN was an ideal choice for several reasons. First, its source code is freely available. More importantly, however, it provided a standard implementation of the multi-layer perceptron ANN that can be used for posture/activity classification. It also had full support for fixed-point values, which was critical in enabling classification in an environment where floating-point coprocessors were not present in many hardware solutions (mobile computing). Finally, FANN's C-based API made it fairly easy to integrate into existing C#-based architecture.

Training of neural networks for both subjects was an offline process that took place on a PC. First, raw sensor data was presented to a set of Matlab scripts, with the data for each posture provided in a separate file. These data were mined for features and the generated features stored in a per-time unit feature vector. The feature vector collection for each was randomly shuffled and partitioned in order to create the training and validation sets for each posture. Next, a training data file and a validation data file were created by printing each training or validation set to the proper output file. Output files were in FANN training format, with inputs and outputs as floating point values. Finally, a shell script called executable wrappers around FANN library functionality [9] in order to create fixed-point versions of the training and validation data files and to actually train and verify a neural network. The result was a trained neural network and a log file describing the mean-squared error of the network for the provided training and validation data. A file representing this trained neural network was made available to the mobile posture classification application. When run, the application presented per-time-unit data to a feature vector generation function with natural semantics comparable to those of the PC-based vector generation utility. The generated feature vectors were then presented to the neural network described by the file from training, which performed the actual classification.

### 3.3. Feature Selection

Feature selection was performed on the dataset described above using a forward selection process. In each round of selection, a set of reduced feature vectors was chosen from the set of original features. The feature vectors were then split into training (three-fourths) and validation (one-fourth) segments. A new ANN was trained to accept each type of reduced feature vector using the training segments. Next, each validation vector was presented to the proper newly-trained network and the networks' classification of each validation vector was recorded. From these classifications, the error rate of each network was determined by dividing the number of misclassifications that it produced by the number of classifications that it performed. In subsequent rounds, the reduced vector with the lowest error rate was entered directly in the selection processes. Rather, the reduced vectors for all other feature types were the union of the previous reduced vector for that feature type and all previous lowest-error-rate reduced vectors.

### 3.4. Calculations of Accuracies

Four-fold holdout cross-validation was used for estimation of accuracy. Collected data was split into four equally-sized partitions. Each partition was then placed into either the training set or the validation set. Three-fourths of the data formed the basis of the training set. The remaining fourth was used to create the

validation set. The training and verification sets were distinct. All reported accuracies are based only on the validation set.

### 3.5. Benchmarking

Benchmarking was performed by timing each classification algorithm while it performed classification on a set of sample data 250 times. However, when we began the actual benchmark procedure, it became clear that the SVM-based classification model was too large to fit entirely within the memory of our mobile phone-based benchmarking platform. Therefore, we were forced to use a simulation that performs the same number of computations as an SVM with the same number of support vectors. Another difference is that the SVM described in [5] performs recognition on unprocessed (raw) high-dimensional sensor data while the methodology proposed herein need computation of a feature vector. The feature vector computation was factored into the total classification time for ANN. Because both classification methods perform a constant number of operations for an input vector of a given size, no significance was attributed to the contents of the input vector.

## 4. Results

### 4.1. Feature Selection

Sixteen rounds of feature selection were performed. The error rate (defined as the number of incorrect classifications divided by the total number of classifications) for the top feature in each round can be seen in Fig. 3. The rate of misclassification fell for the first four rounds. It remained constant for the next four rounds, before beginning to increase. Thus, the top performers in the first four rounds were concluded to make up the optimized feature vector. These were the raw data from accelerometers two and three, the raw data from weight sensor two, and the standard deviation of the value of accelerometer two. By including only the selected features in the input feature vector, the vector's size can be reduced to include only 61 features ( $\alpha_2, \alpha_3, \omega_{21}$ , and  $\sigma(\alpha_2)$ ).

### 4.2. Performance

The SVM-based classification method took 336,964 milliseconds to perform 250 classifications. The time for performing 250 feature vector computations for ANN use was 120 milliseconds for the full vector and only 5 milliseconds for the optimized vector. Performing the same number of ANN classifications took 41 milliseconds for the full vector and 50 milliseconds for the optimized vector. Since an ANN-based classification can only be performed on a computed feature vector, we must add the times for feature vector computation and ANN-based classification together in order to make a comparison to the SVM based classification method. When this is done, we see that an ANN-based classification with the full vector took  $(120 + 41)/336,964 = 161/336,964 \approx 1/2,100$  as long as SVM-based classification in this test. An ANN-based classification with the optimized vector took only  $(5 + 50) / 336,964 = 55/336,964 \approx 1/6,127$  as long as an SVM-based classification.

### 4.3. Accuracy

Confusion matrices for both full and optimized feature vectors have been included (Tables II and III). The rate of successful classification with the full feature vector was 96.6 percent, while the rate for the optimized vector was 95.2 percent which is comparable to previously published results.

## 5. Discussion

The ANN-based classification method managed to offer a three-orders-of-magnitude speed improvement over the SVM-based classification. Using the optimized vector offered another 3 times performance benefit. This should enable real-time recognition of the postures and activities of interest on a mobile device. There was some loss of accuracy (about 4%) over the SVM-based classification method [5]. An inspection of

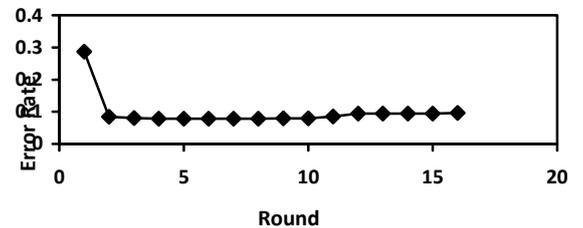


Fig. 3: Error Rate vs. Round Number

confusion matrices reveals a possible reason. Most of errors happen confusing “sit” for “stand” and vice versa. Unlike [5] the sensor system used in this study is a single shoe, therefore shifting the weight on and off instrumented foot may lead to such misclassifications.

Overall, the fixed point ANN offered a robust and computationally light weight mechanism for posture and activity classification.

Table 1: Confusion matrix for full vector

		Predicted Class		
		Sit	Stand	Walk
Actual Class	Sit	1139	57	0
	Stand	46	1142	0
	Walk	19	25	1182

Table 2: Confusion matrix for optimized vector

		Predicted Class		
		Sit	Stand	Walk
Actual Class	Sit	1196	0	0
	Stand	164	1025	1
	Walk	7	3	1218

## 6. Conclusion

We have presented an analysis of the effectiveness of a new method of performing automatic posture recognition using our shoe-based sensor system which is based on Artificial Neural Networks (ANN). We also compare this new method with our previous method, based on SVM. The proposed methodology takes significantly less time to perform than its predecessor with only a small cost to recognition rate. The reduced computational cost of the ANN-based method should enable new battery-saving techniques in the future.

## 7. References

- [1] C. L. Ogden, K. M. Flegal, and M. D. Carroll, “Prevalence and trends in overweight among us children and adolescents, 1999-2000,” *JAMA*, vol. 288, no. 14, pp. 1728 -1732, 2002.
- [2] J. A. Levine, L. M. Lanningham-Foster, S. K. McCrady, A. C. Krizan, L. R. Olson, P. H. Kane, M. D. Jensen, and M. M. Clark, “Interindividual variation in posture allocation: Possible role in human obesity,” *Science*, vol. 307, no. 5709, pp. 584-586, 2005.
- [3] D. M. Bravata, C. Smith-Spangler, V. Sundaram, A. L. Gienger, N. Lin, R. Lewis, C. D. Stave, I. Olkin, and J. R. Sirard, “Using pedometers to increase physical activity and improve health: A systematic review,” *JAMA: Journal of the American Medical Association*, vol. 298, no. 19, pp. 2296 - 2304, 2007.
- [4] W. G. Members, V. L. Roger, A. S. Go, D.M. Lloyd-Jones, E. J. Benjamin, J. D. Berry, W. B. Borden, D.M. Bravata, S. Dai, E. S. Ford, C. S. Fox, H. J. Fullerton, C. Gillespie, S. M. Hailpern, J. A. Heit, V. J. Howard, B. M. Kissela, S. J. Kittner, D. T. Lackland, J. H. Lichtman, L. D. Lisabeth, D.M.Makuc, G.M.Marcus, A.Marelli, D. B.Matchar, C. S.Moy, D.Mozaffarian, M. E.Mussolino, G. Nichol, N. P. Paynter, E. Z. Soliman, P. D. Sorlie, N. Sotoodehnia, T. N. Turan, S. S. Virani, N. D. Wong, D. Woo, and M. B. Turner, “Heart disease and stroke statistics2012 update,” *Circulation*, vol. 125, no. 1, pp. e2 {e220, Jan. 2012. [Online]. Available: <http://dx.doi.org/10.1161/CIR.0b013e31823ac046>
- [5] E. S. Sazonov, G. Fulk, J. Hill, Y. Schutz, and R. Browning, “Monitoring of posture allocations and activities by a shoe-based wearable sensor,” *IEEE: Transactions On Biomedical Engineering*, vol. 48, no. 4, pp. 983 - 990, 2011.
- [6] D. Fulk, George and E. Sazonov, “Using sensors to measure activity in people with stroke,” *Topics in Stroke Rehabilitation*, vol. 18, no. 6, pp. 746 - 757, 2011.
- [7] P. Lopez-Meyer, G. Fulk, and E. Sazonov, “Automatic detection of temporal gait parameters in poststroke individuals,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 4, pp. 594 - 601, 2011.
- [8] N. Sazonova, R. C. Browning, and E. Sazonov, “Accurate prediction of energy expenditure using a shoe-based activity monitor,” *Medicine & Science in Sports & Exercise*, vol. 43, no. 7, pp. 1312 - 1321, 2011.
- [9] S. Nissen, “Large scale reinforcement learning using q-sarsa() and cascading neural networks,” Master's thesis, University of Copenhagen, 2007.