Human Activity Recognition: Accelerometers Unveil Your Actions

Yinglan Ma, Zhongjie Li, Yunong Jiang

1 Introduction

Wearable devices make technology pervasive by bringing it into our daily lives. Apple Watch and smart glasses introduce a new lifestyle while smart bands and wearable medical devices change our ways of keeping track of health. The interaction between our body and the wearable device makes the technology feasible. Thus, it is critical for the device to understand human actions.

Wearable devices become more popular and are usually paired with smartphones. In this work, we explore approaches of recognizing human activities using accelerometer data from smartphones and wearable devices. The dataset we use records acceleration signals from four positions that are representable for smartphones and wearable devices. In real world situation, it is more likely that one only takes one smartphone and another wearable device with him/her. So our ultimate goal is to accurately recognize human actions using two accelerometers. We apply machine learning algorithms to train and infer human motion. The input of our algorithm is acceleration data from sensors. We then use GDA and SVM to output a predicted activity class. The experimental results demonstrate the effectiveness of our method.

2 Related Work

Zhang et al.[12] and Chen et al.[5] studied physics based algorithms for human action recognition (HAR) with acceleration sensors. Physics based method can achieve high accuracy but it's only suitable for small amount of classes. For multiclass classification, machine learning based methods can be conveniently implemented to deal with very complex scenarios. Gao et al. [6] implemented Naive Bayes model in the HAR system but the accuracy was not satisfactory. Ugulino et al.[11] used the AdaBoost ensemble method to classify body postures and movements. Maekawa and Watanabe[8] developed an unsupervised activity recognition with Hidden Markov Models (HMM). Mannini and Sabatini[9] used supervised activity recognition algorithm with HMM. HMM based methods have relatively lower accuracies comparing to Support Vector Machine (SVM). SVM method is widely used in HAR due to the fact that it can classify multiclass datasets with high accuracy conveniently. Davide et al.[2][3] proposed a SVM based methods for the multiclass classification of human motion. He combined fixed-point arithmetic with SVM for computer cost reduction and energy efficiency.

Current state-of-the-art methods are developed by Min et al.[10] using SVM and Naive Bayes, and Ioana-Iuliana and Rodica-Elena[7] using Neural Networks. With data collected from 5 sensors on forehead, both arms and both wrists, Min got 99.4% accuracy. Ioana-Iuliana and Rodica-Elena's method was based on sensor data from right shank and right part of the hip, and achieved an accuracy of 99.6%.

In our work, we plan to explore methods that work well using two sensors, one on waist and the other on common wearable device position.

Motion	Sitting	Sitdown	Standing	Standup	Walking
Occurrences	50631	11827	47370	12415	43390

Table 1: Training and Testing Error: scaled vs unscaled

3 Data

The data used is from UCI machine learning database[1], which captures accelerometer data from four positions, waist, left thigh, right ankle and right upper arm. These four positions represent the common locations of our smartphone and wearable devices, waist for smartphone, and the other three for wearable devices. Each accelerometer reading includes the acceleration signals in x, y and z direction. The dataset also includes user's name, gender, age, height, weight and body mass index. However, these features are less relevant to our problem, so we do not include them as the input. The raw data is sequentially sampled from the accelerometers, so we randomized the data before training and testing. We have a total of 165,633 samples, in which we take 100,000 as training data, 40,000 as testing data. The rest data is used as an extra test set to avoid overfitting the C parameter in SVM. Figure 1 gives an example of raw data from our dataset. Table 1 illustrates the occurrences of five motion classes.

x1	y1	z1	x2	y2	z2	x3	y3	z3	x4	y4	z4	class
-55	129	-139	-480	-502	-603	22	121	-87	-185	-76	-158	walking
-3	92	-63	-23	18	-19	5	104	-92	-150	-103	-147	sitting

Figure 1: Raw Data

3.1 Data Scaling

The data includes acceleration signals at different body positions, in three directions. So the features are on different scale. To make feature values comparable, we test two scaling methods to process our data. Zscore: $f(x_i) = \frac{x_i - \mu}{\sigma}$, and 0-1 Normalization: $g(x_i) = \frac{x_i - x_{min}}{x_{max} - x_{min}}$, where μ and σ represent the average and standard deviation of data x. x_{max} and x_{min} represent the maximum and minimum values of data x.

3.2 Principal Component Analysis

Since the feature dimension of the dataset is 12, it is hard to visualize the data. We use Principal Component Analysis to reduce the dimension to 3, i.e. three principal eigenvectors and eigenvalues are used. Zscore data is visualized as in Figure 2.

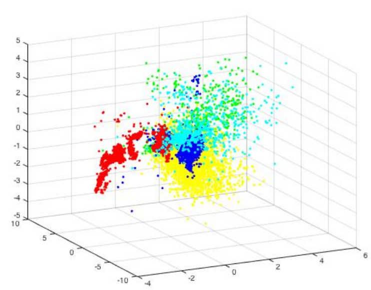


Figure 2: 10,000 Zcore data visualization by PCA

4 Methods

4.1 Gaussian Discriminant Analysis

Since our input features x are continuous-valued random variables, we first try using Gaussian Discriminant Analysis(GDA) model for classification. GDA models $p(x|y)$ as a multi-variant normal distribution, where y is a Bernoulli random variable. We construct a binary classification model for each class $Y = 0, 1, 2, 3, 4$, by assigning $y^{(i)} = Y$ to positive class, and $y^{(i)} \neq Y$ to negative class. Different mean vectors μ_0, μ_1 and one covariance matrix Σ are used.

The model parameters are calculated by the following formulas:

$$\mu_0 = \frac{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 0\}x^{(i)}}{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 0\}} \quad \mu_1 = \frac{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 1\}x^{(i)}}{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\} \quad \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$$

Using one covariance matrix gives us straight line boundary. If we use two different covariance matrices, we will get quadratic boundary. The Σ's are calculated as follows:

$$\Sigma_0 = \frac{\sum\limits_{y^{(i)}=0} (x^{(i)} - \mu_0)(x^{(i)} - \mu_0)^T}{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 0\}} \quad \Sigma_1 = \frac{\sum\limits_{y^{(i)}=1} (x^{(i)} - \mu_1)(x^{(i)} - \mu_1)^T}{\sum\limits_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

When given a new input x, we classify it to $h_\theta(x) = argmax_y \; p(x|y)p(y)$, where $y = 0, 1, 2, 3, 4$.

4.2 Support Vector Machine

A support vector machine is a discriminant classifier defined by separating hyperplane. It aims to find the hyperplane that gives the largest minimum distance to the training examples. Formally, it tries to optimize the following problem:

$$min_{\gamma,w,b} \; \frac{1}{2}||w||^2 \; + \; C\sum_{i=1}^{m}\xi_i$$

$$s.t. \; y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \; i = 1, ...m$$
$$\xi_i \geq 0, \; i = 1, ...m$$

By introducing the $C\Sigma_{i=1}^{m}\xi_i$ term, we allow an example to have margin less than 1, but at a cost of increasing the objective function by $C\xi_i$. C controls the balance between our two goals of making $||w||^2$ small and of ensuring that examples have margin less than 1.

5 Experiments and Results

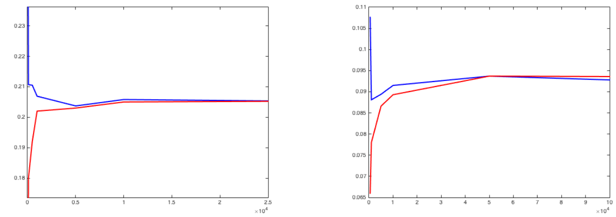
5.1 Gaussian Discriminant Analysis

5.1.1 Single covariance matrix

Using 100,000 training data, and testing on 40,000, we get 79.29% training accuracy and 79.59% testing accuracy. In Figure 3(a), we plot the training error (red line) and testing error (blue line) vs training data size figure. The plot shows little gap between training and testing error, and both of the errors are higher than our desired performance, so the model is highly biased.

5.1.2 Different covariance matrix

Using single covariance matrix gives us a linear decision boundary that underfits our data, so we then explore on using two covariance matrices, which can give us a quadratic decision boundary to fix the underfitting problem. We now get 90.64% training accuracy and 90.72% testing accuracy. The performance has been largely increased. As shown in Figure 3(b), the improved GDA still underfits our data. We also try scaling the data. The results are shown in Table 2. Scaling the data doesn't affect much on the performance.



(a) (b)

Figure 3: (a) Error vs. Training Data Size of single covariance matrix, (b) Error vs. Training Data Size of different covariance matrices. Red line: training error. Blue line: testing error.

	Unscaled	0-1 Normalized	Zscore
Training Accuracy %	90.64	90.64	90.66
Testing Accuracy %	90.72	90.87	90.71

Table 2: Training and Testing Error: Scaled vs Unscaled

5.2 Support Vector Machine

We use LIBSVM [4] to perform SVM on our data.

5.2.1 All Four Sensors

Before scaling our data, SVM performs poorly on all the kernels we have tried, sigmoid, radial basis function(RBF)[1] and 4th degree polynomial kernel.

After using zscore, the lowest training and testing errors we get are around 5%, using RBF. Since the training and testing errors are close, but still higher than our desired performance, the model suffers from high bias. To solve this problem, we increase parameter C. The relation between errors and C is shown in Figure 4.

Training on 100,000 data, and testing on 40,000, we get best performance on SVM model using RBF kernel, with C=1000. We also use extra test set to verify that we are not overfitting C. The training and testing accuracies on each class are shown in Table 3.

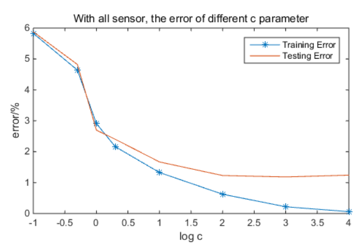


Figure 4: (a) Error vs. logC, using RBF kernel

5.2.2 Two Sensors

We combine the smartphone position(waist) accelerometer data and another wearable device position(left thigh/right ankle/right upper arm) accelerometer data as our input in our next step. As shown in Figure 5(a), when C is small, the training and testing errors are close to each other, but higher than our desired performance. So we are underfitting the data. We then increase our parameter C and get best performance when C=2000. The results are presented in Table 4, 5 and 6, with RBF kernel.

Figure 5(b) illustrates the confusion matrix. Our model tends to confuse between sitting down and standing up. This is likely due to the fact that our data has less occurrences of sitting down and standing up. However, in practice, if we can incorporate the previous motion state we classify into the model, we will do better in distinguishing these two classes. For instance, our model is not sure

[1]Radial Basis Function: $exp(-\gamma|u-v|^2)$. We tried tuning γ, and the default $\gamma = 1$ worked well enough.

	Sitting	Sitting Down	Standing	Standing Up	Walking	All
Training Accuracy %	100.00	99.61	99.85	99.54	99.56	99.78
Testing Accuracy %	99.86	96.69	99.38	96.62	98.19	98.81

Table 3: All Sensors: Error of each motion class

	Sitting	Sitting Down	Standing	Standing Up	Walking	All
Training Accuracy %	99.81	89.53	99.24	83.90	94.84	96.41
Testing Accuracy %	99.76	88.24	99.00	82.53	94.37	96.04

Table 4: Waist+Left Thigh Sensors: Error of each motion class

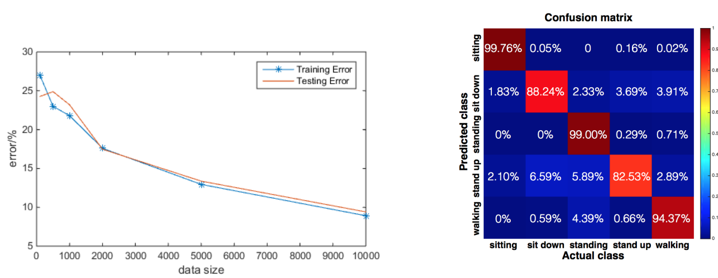
	Sitting	Sitting Down	Standing	Standing Up	Walking	All
Training Accuracy %	99.66	90.75	98.16	77.39	90.24	94.44
Testing Accuracy %	99.51	87.74	98.04	76.62	89.26	93.90

Table 5: Waist+Right Upper Arm Thigh Sensors: Error of each motion class

	Sitting	Sitting Down	Standing	Standing Up	Walking	All
Training Accuracy %	99.64	82.07	98.51	72.42	93.02	94.23
Testing Accuracy %	99.65	80.30	98.48	70.68	92.35	93.88

Table 6: Waist+Ankle Sensors: Error of each motion class

what the current action is, but if the previous action is classified as sitting, then the current state will be more likely to be standing up than sitting down.



(a) (b)

Figure 5: (a) Error vs. Training Data Size of C=0.1 SVM, (b) Confusion Matrix of motion classes

6 Conclusion and Future Work

In this work, we explored how we can recognize people's activities using acceleration data, and we achieved an accuracy of 96.04% in two sensors case, using SVM. SVM casts our classification problem into a higher dimensional space, and it becomes more separable than in a lower dimensional space. The effectiveness of our methods shows that we can accurately classify human actions from two accelerometers: one on smartphone position (waist) and the other on common wearable device position.

For future works, we want to incorporate previous and next action classification into current action recognition. For example, if the previous action is confidently recognized as standing, then we can put a higher weight on the next action of being sitting down and a lower weight of being standing up. Another future work is adding more activity classes, for instance, running, biking and jumping. For more complicated actions, it will be helpful if we can incorporate data from smartphone gyro sensors.

7 References

References

[1] UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets/

`Wearable+Computing%3A+Classification+of+Body+Postures+and+Movements+%28PUC-Rio%29`. Accessed: 2015-09-30.

[2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient assisted living and home care*, pages 216–223. Springer, 2012.

[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *J. UCS*, 19(9):1295–1314, 2013.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] Min Chen, Sanghoon Chu, Jifang Tian, Zhongjie Li, and Hien T Chu. Fall detection scheme using ffs, July 7 2015. US Patent 9,076,471.

[6] Lei Gao, Alan K Bourke, and John Nelson. A system for activity recognition using multi-sensor fusion. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 7869–7872. IEEE, 2011.

[7] F. Ioana-Iuliana and D. Rodica-Elena. Detection of daily movements from data collected with two tri-axial accelerometers. In *Telecommunications and Signal Processing (TSP), 2011 34th International Conference on*, pages 376–380, Aug 2011.

[8] Takuya Maekawa and Shinji Watanabe. Unsupervised activity recognition with user's physical characteristics data. In *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, pages 89–96. IEEE, 2011.

[9] Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[10] Jun-Ki Min and Sung-Bae Cho. Activity recognition based on wearable sensors using selection/fusion hybrid ensemble. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1319–1324. IEEE, 2011.

[11] Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: accelerometers data classification of body postures and movements. In *Advances in Artificial Intelligence-SBIA 2012*, pages 52–61. Springer, 2012.

[12] Yuting Zhang, Stacey Markovic, Inbal Sapir, Robert C Wagenaar, and Thomas DC Little. Continuous functional activity monitoring based on wearable tri-axial accelerometer and gyroscope. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pages 370–373. IEEE, 2011.