Modeling Activity Recognition Using Physiological Data Collected from Wearable Technology

Cezanne Camacho, Jennifer Li, Jeffrey Yang
CS229 Final Project, Stanford University

Abstract

Wearable technology presents a uniquely convenient and portable way to record physiological data from users, which could be used to monitor health or recreational activities. With increasing amounts of such data, it would be useful to automatically categorize a user's activity based on this data. Our paper utilizes machine learning to classify user activity, and we compare the strengths and weaknesses of supervised and unsupervised learning approaches using LDA, SVM and Random Forest classifiers, and K-means clustering classifiers, respectively. We then discuss which of these algorithms show the best performance for general activity recognition.

1. Introduction

As wearable tech becomes increasingly prevalent, vast amounts of additional data will be generated and made available to better understand the activities users are performing in real-time. Wearable tech is also uniquely convenient because of its ability to receive information from an individual user in essentially real time without the need for external infrastructure. Using this knowledge, targeted marketing or predictions can be made about what a user might want to do next based on mood associated with physiological markers. The task at hand is to understand the relationship between the biometric data collected from wearable technology and the activities users are engaged in.

In this paper, we discuss the use of different machine learning techniques for determining a user's activity from a dataset that was collected for the Physiological Data Modeling Contest at ICML in 2004. This dataset maps participant characteristics, such as age and gender, to physiological data that was collected over time during a known activity. Using a training set annotated with codes indicating the type of activity the user was engaged in during the measurements, we train multiclass classifiers to identify the type of activity the participant was performing based on physiological markers. The scope of this study follows the guidelines of the PDMC contest which focused on the ability to distinguish between sleeping, watching TV, and all other activities.

2. Review of Previous Work

Wearable sensor technology has been investigated as an effective way to regularly monitor individual health. Researchers at MIT have used wearable technology to build a mobile, personal profile that records vital signs, motor activity, and sleep patterns so that users can look at these health indicators on a real time basis [1], and some research has been done to use machine learning to map emotional state to features from physiological data [2]. Wearable accelerometers have been used to quantify and classify motor ability in recovering stroke victims [3], and accelerometers have been used in elderly care to recognize when someone has fallen [4].

Especially in these health-related applications, it is easy to see that any activity classification algorithms must be very accurate and robust in the face of a changing user. Most machine learning applications rely on learning from heavily annotated data run through SVM classifiers, and since data is often not so readily categorized, our project aims to see how unsupervised learning algorithms compare to supervised approaches.

3. Dataset, Features, and Preprocessing

We obtained our data from a set that was collected for the Physiological Data Modeling Contest at ICML in 2004. This dataset was collected from participants wearing BodyMedia wearable technology and includes details of participant age, handedness, and gender, as well as physiological markers including galvanic skin response, heat flux, body temperature, skin temperature, and accelerometer measurements.

The physiological data that each sensor records is specified in Table 1. Data was collected from 18 users across several sessions, with measurements taken each minute. Over 700k training examples were made, of which approximately 200k were annotated.

Table 1. Semantics of the characteristics of the human subjects and the sensor readings

Name	Semantics
characteristic1	age
characteristic2	handedness
sensor1	gsr low average
sensor2	heat flux high average
sensor3	near body temp average
sensor4	pedometer
sensor5	skin temp average
sensor6	longitudinal accelerometer SAD
sensor7	longitudinal accelerometer average
sensor8	transverse accelerometer SAD
sensor9	transverse accelerometer average

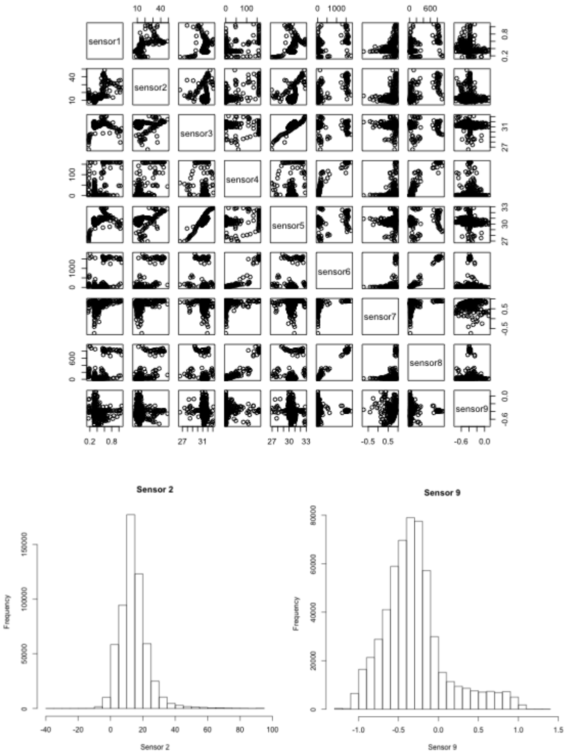


Figure 1. Correlations between biometric sensors and select training data distributions.

Figure 1 shows the correlations for each of the nine sensors for which we can see that the physiological data are not always independent of one another. Indeed, features like average skin temperature and heat flux are naturally correlated, which is accounted for during feature selection for some of our algorithms. Figure 1 also shows sample data distributions which were transformed as-needed during pre-processing.

The data was pre-processed by first removing outliers and incomplete entries. Transformations were applied to reduce spread and skew in the data. New binary features, such as a 'walking' field from pedometer data, were created to use in model training. Activity annotations were reclassified into either sleeping, watching TV, or other, as outlined in the PDMC contest.

4. Implemented Algorithms

We aimed to compare the efficacy of supervised versus unsupervised learning algorithms by implementing linear discriminant analysis (LDA), support vector machines (SVM) with different kernels, and the random forest model for supervised models, and a K-means clustering approach for the unsupervised model. Here, we discuss the method behind each approach and the results that were produced.

4. 1. Linear Discriminant Analysis (LDA)

Linear discriminant analysis reduces the feature dimension, and separates data points into classes based on the reduced feature subspace. This reduced features space is computed through maximizing the separation between multiple classes. Because the algorithm requires this information about the classes, LDA is a supervised learning algorithm. The main steps of the algorithm start with computing the means of the feature vectors and computing the within-class scatter matrix, S_W, using these means. The between class scatter matrix, S_B, is then computed using the sample sizes and means of all the classes in the training data. To incorporate the effects from both within and between class variations, the algorithm computes the eigenvalues for the matrix, $S_W^{-1}S_B$. The eigenvalues play an important role in determining the new features subspace, as the eigenvalues with lower values contain less information about the distribution of the data. A number of eigenvalues are chosen to construct the new features space, and a matrix, W, is computed to transform the samples into the new subspace. The final objective function, D, for the algorithm is to maximize:

$$D(W) = \left|\frac{W^T S_B W}{W^T S_W W}\right|$$

For our classification of activities, we need to distinguish between three classes, sleep, TV, and other. We use LDA to reduce the number of features to 2, and classify the activities based on this new subspace, as shown in Figure 2 below. The confusion matrix, as well as the Precision, Recall, and F-scores for this algorithm are shown in Table 2.

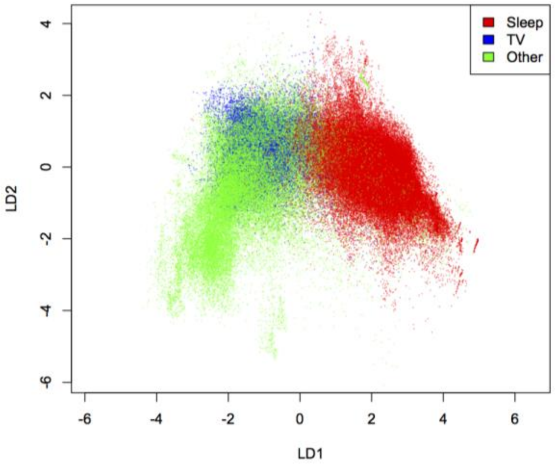


Figure 2. Classified data plotted on the reduced feature subspace determined through LDA showing optimal separation between sleep, TV and other.

Table 2. Confusion matrix and Precision, Recall and F-scores for LDA model.

	Other	Sleep	TV
Other	55170	2029	20914
Sleep	11206	101196	2117
TV	3768	441	3100

	Precision	Recall	F-Score
Other	0.706	0.787	0.744
Sleep	0.884	0.976	0.928
TV	0.424	0.119	0.185

4.2. Support Vector Machines (SVM)

Support Vector Machines is a supervised learning algorithm that classifies objects based on the support vectors of a dataset, or the data points that lie closest to the decision boundary. The algorithm aims to maximize this distance to better separate the classes. The objective function that achieves this is the following:

$$\max D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i \alpha_i y_j \alpha_j K(\boldsymbol{x}_i x_j)$$

$$subject\ to \begin{cases} \forall_i \; 0 \leq \alpha_i \leq C \\ \Sigma_i \; y_i \alpha_i = 0 \end{cases}$$

This function arises from the primal form, which maximizes the functional margin, or essentially how far the closest sample point is from the boundary. To make this problem easier to solve, we work with the dual form of the objective, shown above. In most cases, α will be zero. In the few cases of the support vectors, the α will be nonzero. The kernel in the equation allows for feature mapping and nonlinear decision boundaries.

Initially, we trained SVMs using 10% of the training data using the default parameters in the R package e1071 and altered the kernel function type (linear, radial, polynomial and sigmoidal) in order to determine which kernel provided optimal classification. The radial kernel function was found to provide the best performance, and the sigmoidal kernel the worst, which is highlighted in Table 3. Using the radial kernel, training and test errors were measured as a function of the number of training examples, ranging from 17k to 170k examples (Figure 3). Increasing the number of training examples resulted in small improvements in both reducing measured training and test errors. This suggests that we should further optimize our choice of features rather than increase our training set size. The confusion matrix and Precision, Recall, and F-scores are shown for the radial kernel in Table 4. Based on these results, we can conclude that our SVM model provides more accurate classification than LDA, with almost a doubled F-score for the TV class, which is the most frequently misclassified activity.

Table 3. Comparison of SVM training and test errors using various kernel functions.

Kernel Type	Training Error	Test Error
Linear	0.1627	0.1981
Radial	0.1354	0.1965
Polynomial	0.1481	0.2050
Sigmoidal	0.2185	0.2636

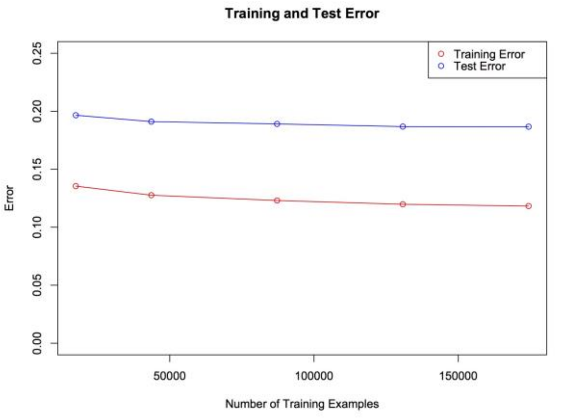


Figure 3. Training and test error of SVM with radial kernel function as a function of training examples used in model fitting.

Table 4. Confusion matrix and Precision, Recall and F-scores for SVM with radial kernel.

	Other	Sleep	TV
Other	55412	2870	18034
Sleep	10127	100676	1567
TV	4605	120	6530

	Precision	Recall	F-Score
Other	0.726	0.790	0.757
Sleep	0.896	0.971	0.932
TV	0.580	0.250	0.349

Table 5. Confusion matrix and Precision, Recall and F-scores for Random Forest model.

	Other	Sleep	TV
Other	56907	3514	17723
Sleep	8552	100039	1494
TV	4685	113	6914

	Precision	Recall	F-Score
Other	0.728	0.811	0.768
Sleep	0.909	0.965	0.936
TV	0.590	0.265	0.365

4.3. Random Forest

Random Forest fits decision trees to randomly selected samples of data from the training set and features, and makes predictions for new data by averaging the predictions from all trees. We applied a random forest model of 1000 trees, which showed slightly improved performance compared to our SVM model. Improved F-scores were measured for all three classes of activity compared to our SVM model. The confusion matrix and Precision, Recall, and F-scores are shown in Table 5.

4.4. K-Means Clustering

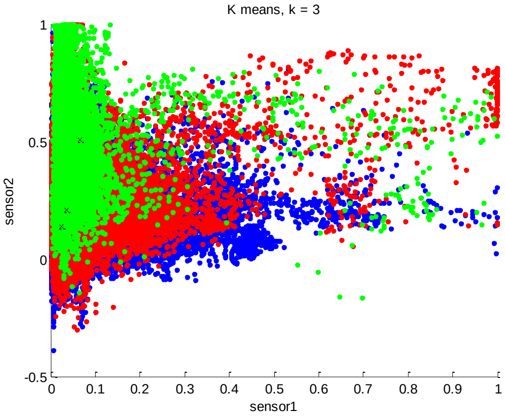
K-means is an unsupervised algorithm that clusters data based on how close they are to a determined cluster centroid. It essentially relies on the closeness of data that describes one class, and distance between separate classes of data. We thought that k-means applied to activity recognition would be useful to compare to supervised learning algorithms as well as useful because it has the advantage of not requiring annotated data, which for wearable technology would make data collection easier.

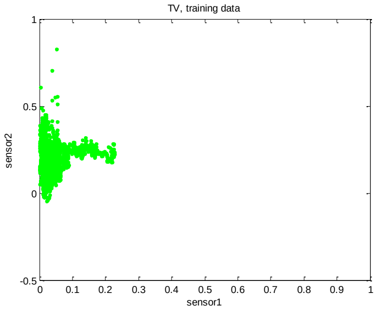
In order to apply K-means to activity recognition based on sensor data, we take the set of training data and sample three sensor data examples randomly to create three initial k means, which are also our initial centroids. These are three vectors of length nine to account for each sensor value. We then go through all of the sensor data, $x^{(i)}$ and calculate which of these three means it is closest to as determined by the squared error to create cluster of data points; one cluster for each of the centroids. We then update the value of the centroids based on minimizing the error between the centroid and the cluster sensor data that is assigned to them. This is the objective function of the k means algorithm and is described in the equation below, where D is the function to be minimized, xi the nine-dimensional data points in the training set, and c_j the current centroids.

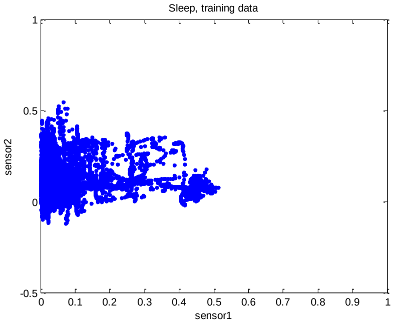
$$D = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

We then run these steps for forming clusters and updating centroids repetitively until the values of the centroids stop changing or have only changed a trivial amount. Then we can say that this algorithm has reached convergence and that a local optimum has been reached. K-means does not guarantee that a global optimum has been reached because that optimum is dependent on the initial random choice of centroids. K-means also works best for clusters of data that are well separated, but in the case of physiological data, there is often a lot of overlap.

We applied K-means to the separation of three activities: sleeping, watching TV, and other. K-means with k=3 was applied to cluster the three activities in a nine-dimensional space. The K-means clusters and centroids are shown in Fig 4, along with the actual class distributions. The error of K-means prediction was 33.79% (taken as an average over three runs).







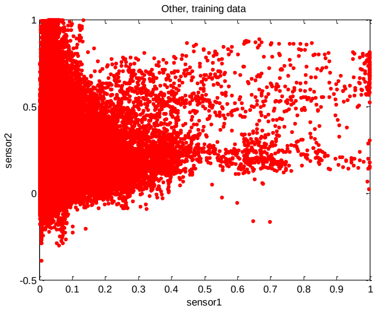


Figure 4. Sample of results of the K-means clustering algorithm for k = 3. (From top to bottom) The first image shows the resultant clusters for one run of K-means; the ‘x’s mark the centroids of the clusters. The next three plots show the actual division of training set data into 3 clusters indicating TV, Sleep, and Other. These images show clusters for sensors 1 and 2 so that they can be plotted in a 2D space, but all nine sensors were used in computing the clusters.

5. Conclusions and Future Work

This project demonstrated how machine learning can be applied to a variety of specific activity-related diagnostics. Our work shows that supervised learning algorithms provide the most accurate recognition, with the Random Forest algorithm having the most promising F-scores, which indicate accuracy based on precision and recall. Our unsupervised algorithm could not separate the data as well as the supervised approaches. However, work could be done to further improve this approach, which would make it easier to handle large amounts of individual data supplied by wearable tech. To improve the accuracy of our models, additional pre-processing could be done, including normalizing the biometric sensors for each individual and training models based on an data that precludes variation across individual participants. Neural networks and time-dependent approaches like Conditional Random Fields could be implemented to provide more accurate predictions using knowledge of previous activity annotations.

One could imagine how activity recognition could be applied to individual healthcare, say fall detection for the elderly or general fitness, as well as how this recognition could be applied to personal entertainment and recreation.

6. References

1. Chieu, H., et. al. “Activity Recognition from Physiological Data Using Conditional Random Fields.” Computer Science MIT. (2006)
2. Pentland, A. “Healthwear: Medical Technology Becomes Wearable.” Computer 37.5 (2004)
3. Picard, R.W., E. Vyzas, and J. Healey. “Toward Machine Emotional Intelligence: Analysis of Affective Physiological State.” IEEE Transactions on Pattern Analysis and Machine Intelligence 23.10 (2001): 1175-191
4. Hughs, R., T. Hester, J. Stein, and S. Patel. “Tracking Motor Recovery in Stroke Survivors Undergoing Rehabilitation Using Wearable Technology.” Engineering in Medicine and Biology (2010)
5. Lee, Y., and M. Lee. "Accelerometer Sensor Module and Fall Detection Monitoring System Based on Wireless Sensor Network for E-Health Applications." Telemedicine and E-Health 14.6 (2008): 587-92