Recognizing and classifying daily human activities through data recorded by embedded inertial sensors

Divyam Gupta
2018032
divyam18032@iiitd.ac.in

Saksham Dhull
2018186
saksham18186@iiitd.ac.in

Vrinda Singhal
2018421
vrinda18421@iiitd.ac.in

Abstract

Human Activity Recognition (HAR) has wide applications in medical research and human survey system. In this project, we design a robust activity recognition system based on a smartphone. The system uses a 3-dimentional smartphone accelerometer and gyroscope to collect time series signals, from which 561 features are generated in both time and frequency domain. The feature space is reduced to 200 using PCA. Activities are classified using 6 different methods, i.e., Random Forest, Gaussian Naive Bayes, Logistic Regression, Multi-layer Perceptron, Support Vector Machine, and Extra-trees. Experiment results show that the classification rate of SVM reaches 98.26 and is robust to common positions and poses of cellphone.

Code Repository: Human-Activity-Recognition on Github

1. Introduction

In the modern digital age, smartphones and wearables are becoming increasingly ubiquitous. Embedded inertial sensors like accelerometers and gyroscopes in these devices pick up a huge number of measurements at high frequency. Human activity recognition (HAR) is a way of comprehensively understanding these measurements to develop key observations regarding a person's lifestyle. HAR has wide-ranging applications in the field of wearable technology and entails classification of regular activities like walking, standing, laying, sitting, etc. through the measurements obtained. Such a classification can be applied in elder care support, rehabilitation assistance and in diagnosing diabetes and cognitive disorders.

2. Literature survey

Human activity recognition is a relatively new field and has an extensive scope for research. Initial work by Rasekh *et al*. [1] uses accelerometer data from an HTC Evo Smartphone and arrives at the highest accuracy using SVMs coupled with active learning techniques. Usharani *et al*. [2] presents a clustered K-Nearest Neighbours algorithm to arrive at an accuracy of 92% using tri-axial motion sensor data. M Badshah [3] compares the use of feature engineered data on traditional models like Random Forest, Decision Tree and ANNs with deep learning techniques using raw sesnor data.

3. Dataset

Anguita *et al*. [4] have published a public domain dataset for human activity recognition using smartphones. We used this dataset for our implementation. The dataset comprises of 3-axial linear acceleration and 3-axial angular velocity measured at a constant rate of 50Hz using the embedded accelerometer and gyroscope of an attached device. After noise reduction and signal processing on the raw inertial data, the time signals are sampled in a sliding window of 2.56 seconds, with a 50% overlap. Moreover, standard measures like mean, median, standard deviation etc. are also calculated.

This yields a total of 10929 samples, which are divided in a 70:30 ratio by keeping individual subjects in either the training or testing set. The activities are classified into six primary classes - walking, sitting, standing, walking up, walking down and laying. Additionally, there are six classes which denote transition states, like sitting to standing, standing to laying, etc.

4. Methodology

4.1. Data Pre-processing

Features were normalized and bounded within [-1,1]. Rows with any missing values were removed and the rows were shuffled. We merged the original train and test data since they had a disproportionate distribution of classes. New training and testing data data was produced with 70-30 stratified split.

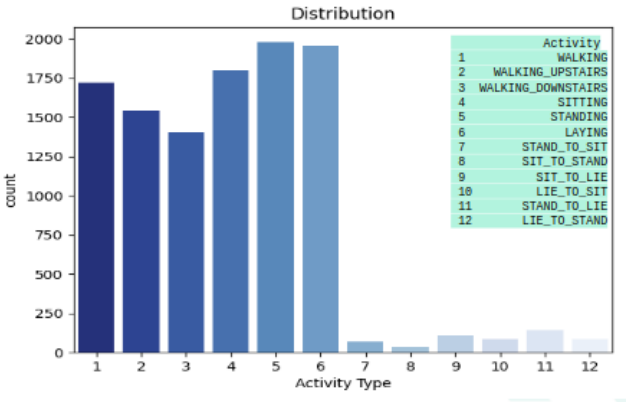

Figure 1. Class Distribution

4.2. Exploratory data analysis

We began preliminary analysis by checking the class distribution in the dataset. From Figure 1. we can observe that the primary classes (from 1-6) are represented well in the dataset while the transition states are relatively low in number.

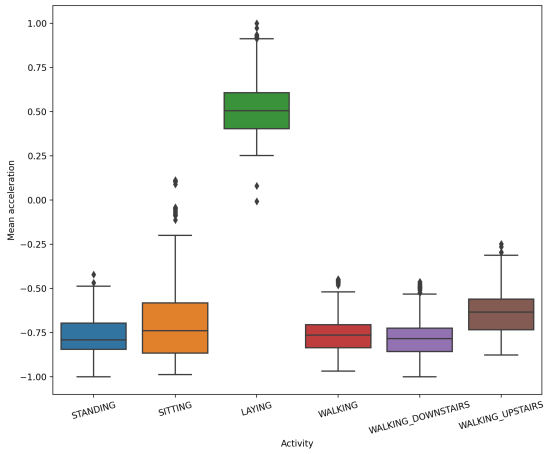

Figure 2. Mean x-axis gravity measurements for various activities

A deeper dive into the dataset revealed greater insights into the data. For example, in Figure 2. we can see that the mean x-axis gravity measurement (using accelerometer data) is the highest when the person is laying. This happens due to the horizontal orientation of the attached device. Thus, this measurement can act as a differentiating feature to classify the activity into the laying category.

Lastly, we observed the deviation in the body acceleration measurements. From Figure 3. we can observe that static activities like sitting, standing and laying have little

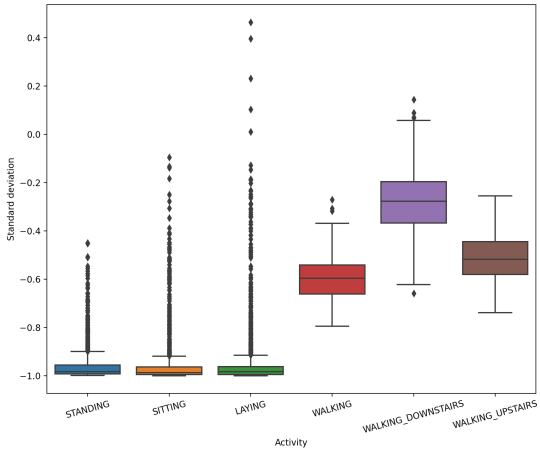

Figure 3. Standard deviation in body acceleration measurements for various activities

to no deviation. On the other hand, the highest deviation in acceleration is observed when the person is walking downstairs.

4.3. Dimensionality Reduction

The original dataset captures the sensor data using 561 features. Training 70% of the dataset, i.e., approximately 7,650 samples would be computationally expensive with the given number of features. Thus, we employed Principal Component Analysis [5] (PCA) as our dimensionality reduction technique. From Figure 4. we can observe that 90% of the variance in the data can be explained using only 200 features, which is a significant improvement over the 561-dimensional feature space available to us initially.

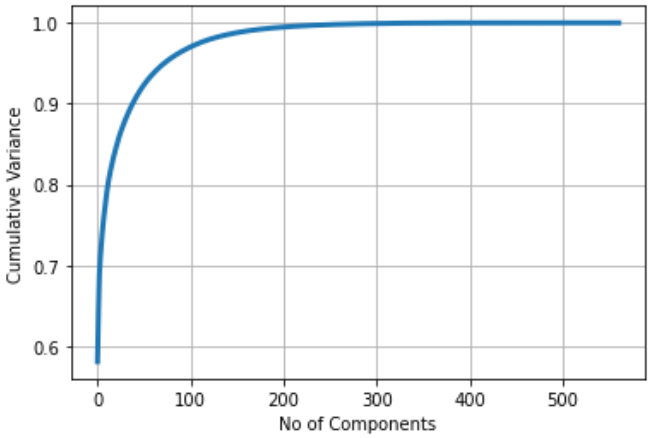

Figure 4. Cumulative explained variance v/s number of components for the training data

To further visualise the data, we used t-SNE [6] (t-Distributed Stochastic Neighbour Embedding) after using PCA. We observed better clustering and accuracy by using t-SNE on the reduced data, as compared to using t-SNE on the higher dimensional data. From Figure 5. we observed that the data has distinct clusters for each class of activity.

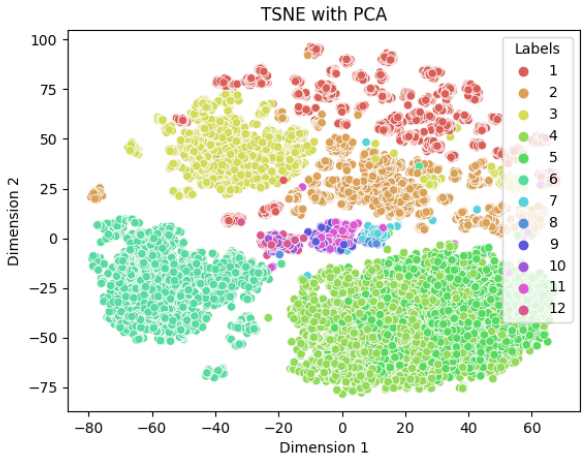

Figure 5. t-SNE clustering after applying PCA on the original data

4.4. Training

We trained the following classifiers and obtained optimal hyperparamters for the best performing model using a grid search on various hyperparameters.

Logistic regression

Logistic regression is a linear regression technique primarily used for binary or discrete response variable. We use scikit-learn's [7] implementation of logistic regression for multi-class classification. It uses the LBFGS (Large-scale Bound-constrained Optimization) solver internally for the multiclass implementation. We also apply L1 and L2 regularization as penalties to avoid overfitting.

$$L1(w) = ||w||_1 \tag{1}$$

$$L2(w) = ||w||_2^2 \tag{2}$$

Gaussian Naive Bayes

The Naive Bayes algorithm is a fast and easy to implement algorithm which can classify multi-class data in an efficient manner. It uses binning techniques to classify continuous data into discrete classes.The underlying principle of this classifier is the Baye's theorem for conditional probability. The Gaussian probability distribution used is defined as:

$$P(x = v|C_k) = \frac{1}{\sigma_k\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma_k^2} \tag{3}$$

A major weakness of the Naive Bayes algorithm is the assumption that the features are independent of each other.

Random Forest

The random forest classifier [8] is an ensemble classification technique. It works on the principle of bagging, i.e., bootstrap aggregation, to generate a certain number of decision trees. Additionally, it randomly selects a subset of all the attributes to calculate information gain or gini impurity at each split, as the case may be.

Using these stochastic methods, it aims at reducing the overfitting problem encountered in decision trees.

Extra-trees classifier

The Extra-trees algorithm [9] constructs a certain number of random trees using the entire training subset provided to it, rather than using a bootstrap replica of the same. A node in the decision tree is split using information gain or gini impurity from a subset of attributes selected at random. Thus, it acts as a truly random ensemble learning method.

Support Vector Machine

Support Vector Machine [10] is a supervised learning technique which can be used for classification and regression tasks. A pre-requisite of the SVM algorithm is that the input data should be linearly separable in case of a classification task. The SVM works by finding the optimal hyperplane which maximises the margin between the datapoints of the two classes.

The optimal hyperplane is defined as:

$$g(x) = w^T x + b \tag{4}$$

where w are the weight vectors which yield the maximum margin. The objective function to be minimised is given by:

$$\phi(x) = \frac{w^T w}{2} \tag{5}$$

In the event that the data is not linearly separable, the SVM algorithm maps the training data into higher dimensions to make it linearly separable. The kernel function of the SVM defines the generalised dot product that is used to map the data into higher dimension.

Multi-layer perceptron

Multi-layer perceptrons [11] are a class of artificial networks. They provide a non linear mapping between the input and output data through a network of hidden perceptrons. These perceptrons successively apply activation functions on the input data and provide corresponding outputs on the output layer. The multi-layer perceptron optimises the synaptic weights between consecutive layers using the backpropagation algorithm. For backpropagation, the algorithm defines a loss function, and the resultant error signal corresponding to each input is propagated backwards through the entire network for weight updation.

We use scikit-learn's [7] implementation of the multi-layer perceptron with two hidden layers of sizes 256 and 128. This implementation considers regularization and Nesterov's momentum [12], which allows for adaptive learning and better generalisation. We used the following activation functions for the network:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{6}$$

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{7}$$

$$ReLU(x) = \max(0, x) \tag{8}$$

$$linear(x) = x \tag{9}$$

5. Results and analysis

We trained various models by keeping the Gaussian Naive Bayes classifier as the baseline model, with a testing accuracy of 83.71 %. Table 1. shows the testing accuracies of different models trained using the original feature space and the PCA reduced feature space. We observed consistent improvements in accuracy, the highest being with the SVM. For the same model, dimenisionality reduction reduced the accuracy for tree-based models like random forest and extra trees classifier. A similar trend was observed for the k-Nearest Neighbours algorithm. Thus, from Table 1. we can infer that SVM is the best classifier for the given sensor data.

5.1. Implications of the results

A comparison of the accuracies as shown in Table 1. shows the weakness of each of the models. We can see that for Gaussian Naive Bayes, dimensionality reduction using PCA increases the accuracy significantly. Gaussian Naive Bayes expects the features to be independent, which is not the case with sensor data. PCA reduces correlation between individual features, which reduces the dependence between the features, thereby increasing the testing accuracy significantly.

The random forest and the extra-trees classifiers perform better than the Naive Bayes algorithm, however, a loss of accuracy is observed when PCA is applied before training. These models score lesser since they try to avoid overfitting to an extent through bagging. However, as we can observe, this leads to lower accuracies in comparison to the multi-layer perceptrons and Support Vector Machines

We ran multiple experiments on the multi-layer perceptron by changing the hidden layer activations, and tuning hyperparameters like the regularisation hyperparamter and the Nesterov momentum. Dimensionality reduction marginally reduces the accuracy of the model, showing that the model is robust to the number of input features.

Logistic regression performs well both on the PCA reduced space and the original data space, with only marginal difference in testing accuracy. The L2 regularization used improves the performance of the model significantly, leading to the high accuracy obtained.

The SVM performs the best among all the models tested. Since the SVM finds the margin-maximising hyper plane using a non-linear kernel, it is able to classify most of the samples correctly. From Figure 5., i.e., the t-SNE plot, we can see that the data is well clustered, which allows for finding a decisive boundary between members of each class.

5.2. Hyperparameter tuning

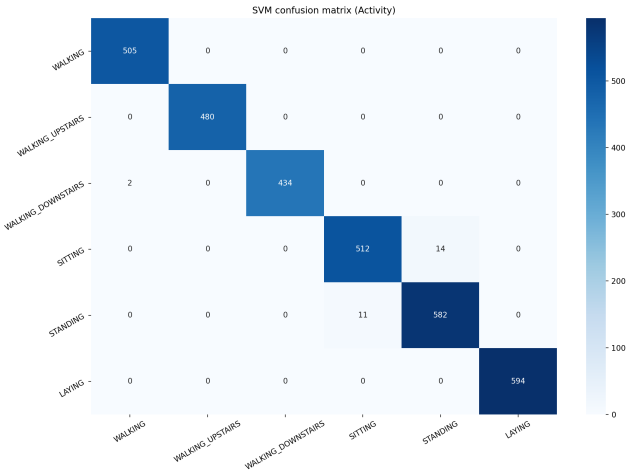

Figure 6. Confusion matrix for the SVM with optimal hyperparameters

To further improve testing accuracy, we ran a grid search on the SVM. The hyperparameters chosen were C, gamma

Model used	With PCA	Without PCA
Gaussian Naive Bayes	83.71%	69.44%
Random Forest classifier	92.23%	96.30%
Extra-trees classifier	93.37%	96.77%
Multi-layer perceptron (with ReLU activation)	96.40%	96.52%
Logistic Regression	97.59%	97.62%
C-Support Vector Classifier	**98.26%**	98.20%

Table 1. Summarisation of accuracies obtained with and without dimensionality reduction

and the kernel. We varied the C and gamma values in the interval $[0.001, 100]$, with a 10 times increase at each step. The kernels chosen for the grid search were the polynomial and RBF kernel.

The optimal hyperparameters obtained are as follows:

C = 100; **gamma** = 0.01; **kernel**= RBF

The above hyperparameters yielded the highest accuracy of 98.26% on the PCA reduced space. Figure 6. shows the confusion matrix for the SVM with the optimal hyperparameters. The confusion matrix is shown only for the orientations/movements that are well reperesented in the data, i.e., walking, walking downstairs, walking upstairs, sitting, standing and laying. The transition movements are excluded in the matrix since they're not well represented in the data. Furthermore, Figure 7. shows the ROC curve for the SVM.

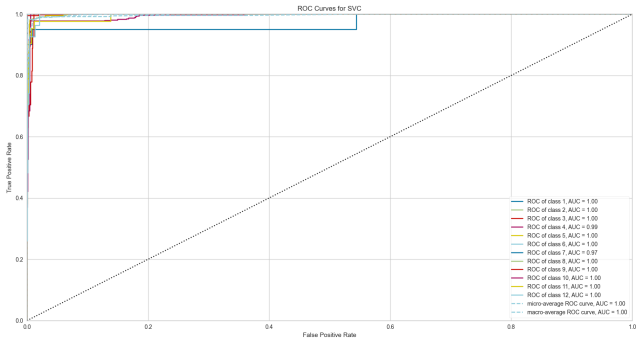

Figure 7. ROC curve for the SVM with optimal hyperparameters

6. Limitations and future scope of work

Resource consumption

Since human activity recognition is a desirable feature of wearables and smartphones, it is yet to be seen the energy and CPU consumption of the models which we trained and tested. Developing much more efficient models would prove to be a breakthrough in the field.

Inclusivity

The dataset which is available consists of movements from volunteers in the common age range of 19-48. This renders our models to be completely user dependent and less inclusive. Including diverse subjects like elderly people, pregnant women, people with disabilities etc. would provide more inclusive and richer insights into the movements of a diverse set of people. This would allow developers to provide personalised recommendations and can potentially change the e-healthcare market.

7. Conclusion

In this project, we learnt how to process the time series signals collected using a built-in accelerometer and gyroscope in time and frequency domain. We reduced the feature dimensionality from 561 to 200 using PCA to improve the performance. The activity data were trained and tested using 6 learning methods. We found that classification performance is robust to the orientation and the position of smartphone. Conclusively, we found SVM to be the optimal choice for our problem with a classification rate of 98.26%.

8. Individual Tasks

All the group members contributed equally to each phase of the project.

1. *Divyam* - Data pre-processing and cleaning, hyperparameter tuning, analysis of results, report writing

2. *Saksham* - Data collection, training classifiers, analysis of results, report writing

3. *Vrinda* - Feature selection, plotting and analysis of results, report writing

We adhered to the timeline diligently and presented our work to the best of our abilities.

9. Acknowledgements

We would like to thank Dr. Jainendra Shukla for conducting the course and guiding us through the various learning techniques in the domain of Machine Learning.

References

[1] Amin Rasekh, Chien-An Chen, and Yan Lu. Human activity recognition using smartphone. *arXiv preprint arXiv:1401.8212*, 2014.

[2] J Usharani and Usha Sakthivel. Human activity recognition using android smartphone. In *Proceedings of the 1st International Conference on Innovations in Computing & Networking (ICICN-16), Mysore Road, Bengalura*, pages 12–13, 2016.

[3] Mustafa Badshah. Sensor-based human activity recognition using smartphones. 2019.

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, page 3, 2013.

[5] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

[8] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[10] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[11] Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.