Human Activity Recognition Using Smartphones Data Set

Moghey, Manasee (Manasee_moghey@student.uml.edu)
Rao, Anuroop G(Anuroop_rao@student.uml.edu)
Sharma, Aanchal(Aanchal_sharma@student.uml.edu)

Abstract

In this project, our goal is to extract useful information from the dataset, apply various classifiers to the chosen features and analyse their performance. Activity-Based Computing [1] captures the state of the user and its environment by exploiting heterogeneous sensors in order to provide adaptation to exogenous computing resources. When these sensors are attached to the subject's body, they permit continuous monitoring of numerous physiological signals. This has appealing use in healthcare applications, e.g. the exploitation of Ambient Intelligence in daily activity monitoring for elderly people.This report discusses the classifiers used and comparison to seek the best performing one. We found that the SVM with linear kernel is able to provide more precise performance.

1. Introduction

The use of smartphones has become an inalienable part of everyday life of people of almost all ages globally. 91% of the world population owns a mobile device.Besides the basic function of telephony, smartphones provides many other features. Out of all the features, multitasking and the deployment of various sensors, are currently merged to the current handsets. It is pictured that with those features, we are capable to use smartphones to keep track of our activities. We can learn from them, and later assist us to make better decisions in future. In this report, we apply various algorithms to analyze those data of accelerometer and gyroscope, and then predict and recognize six human activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying).

The report is structured in the following way: The state of the art regarding previous work is depicted in Section 2. The description of the data set is given in Section 3. The description of the adopted methods and comparison of the analyses results is presented in Section 4. The explanation of the best performance to distinguish between the rest state and other states using SVM with linear kernel classifier is presented in Section 5. The conclusions of this research project are described in Section 6 and lastly, the acknowledgement in Section 7.

2. Background and Related Work

There has been a variety of work done in this field. Very early work tries to identify human activities based on different kinds of data source [5]. Some data comes from sensors like body worn sensors, which may not give much information about human activity [2]. Whereas some data comes from specialized devices, which are costly, uneasy for users and become a headache to the user during regular activity [3]. Thus more and more researches are focusing on using wearable smart devices' data to classify human activity[4]. In this project, we used a dataset obtained from smartphones containing data related to both activities and postural transitions.

Bao et al. [6] developed algorithms to detect bodily activities from everyday tasks, and observed that while some activities are classified more accurately with subject-independent training data, others require subject-specific training data. This suggests that multiple sensors aid in recognition because conjunctions in acceleration feature values can help to identify many activities.

3. Data Set

The dataset includes all triaxial acceleration from the accelerometer and triaxial angular velocity from the gyroscope at a constant rate of 50 Hz captured integrated in the smartphones. We have 561 features. UCI link is:

https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones. The experiments have been carried out with a group of 30 volunteers within an age range of 19-48 years. Each person performed six activities (SITTING, STANDING, LAYING, WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS) wearing a smartphone (Samsung Galaxy S II) on the waist. Features comprises of these six activities in both time and frequency domain along with the corresponding to each data point with 561 features.

4. Approach

There are several algorithms which are capable for multiclass prediction of the dataset. We've used LDA, Logistic Regression, K-Nearest Neighbors and SVM. We compare the confusion matrix of those classification results on the test data using the different training classifiers, presented by confusion matrices.

(i) Linear Discriminant Analysis:

We have used multi-class LDA which is a generalization of two class LDA problem. For this approach we create two kinds of matrices, one is within class and other is between class. The within class matrix can be calculated as:

$$\mathbf{S}_w = \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}_{y_i})(\mathbf{x}_i - \boldsymbol{\mu}_{y_i})^T$$

Here, is the sample mean of the k-th class.

The between class matrix can be computed as:

$$\mathbf{S}_b = \sum_{k=1}^{m} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$

Here, m is the number of classes, is the overall sample mean, and is the number of samples in the k-th class.

To form the multi class LDA we create an optimization problem where we find linear combinations of w that maximizes the ratio between-class to within-class scattering as:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$$

We can get the solution by forming a generalized eigenvalue problem

$$\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w}$$

We implemented LDA from scratch and we used MATLAB for the same.

(ii) Logistic Regression:

The approach here generalizes the logistic regression to a multi-class problem where the outcome is more than two classes. Given a set of independent variables, a multinomial LR is used to predict the probability of different possible outcomes a categorically distributed dependent variable.

For likelihood calculation, we use 1-to-K coding scheme in which we create the target value y for each example, if y matches k then the output vector will have 1 for k-th value and remaining values is 0.

s. The log likelihood function for the ith example is given by

L(w1, w2,..,w_n) = sum (y_i^k log p(k/x_i))

This requires evaluation of the Hessian matrix that comprises blocks of size M × M in which block j, k is given by

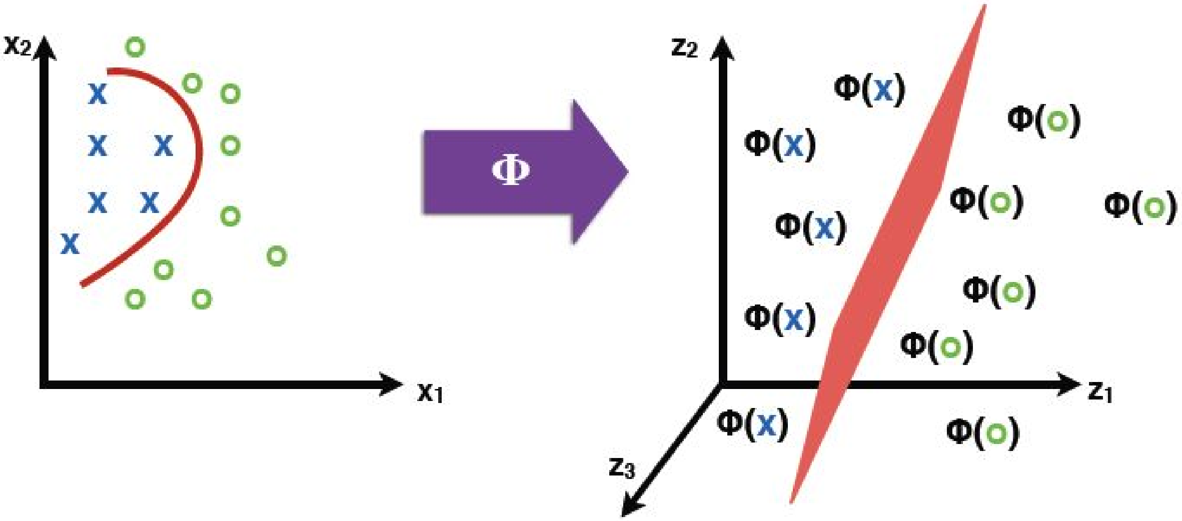
$$\nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N} y_{nk}(I_{kj} - y_{nj}) \phi_n \phi_n^{\mathrm{T}}.$$

As with the two-class problem, the Hessian matrix for the multiclass logistic regression model is positive definite and so the error function again has a unique minimum.

(iii) Support Vector Machines:

SVM are the supervised learning models which helps to examine data used for classification and regression analysis. Suppose we have a set of training examples, each example marked as belonging to one or the other of two categories, SVM designs a model that assigns new examples to one category or the other. The support vector machine is fundamentally a two-class classifier. But we often have to tackle problems involving K > 2 classes. Multiclass SVM assigns labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. With the help of kernels, we can map the input data into high dimensionality feature space, where we can find the linearly separable boundary.

$$x \to \Phi(x)$$



As we know that the main difficulty for binary SVM is on the density of the kernel matrix as in general K(ij) is not zero. The decomposition method is the major method to solve binary SVM. LibSVM is used for the selection of the working set. The Kernel trick is used to avoid high dimensional feature space computations. It helps to formulate algorithm in terms of dot products of feature vectors. Instead of using explicit computations involving feature vectors, we use kernel function for computations. This function computations has x and x' in original input space k (x , x'). Mercer Theorem helps to determine if given function is a valid kernel or not.

Linear kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over linearly separable data.
The following is formula for linear kernel:

$$K(X,Y) = X^T Y$$

The formula for polynomial kernel is given below:

$$K(X,Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

The Radial basis function (RBF) Kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right)$$

The polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of nonlinear models.

(iv) K-Nearest Neighbor:

K-nearest neighbors algorithm (k-NN) is a non-parametric method, which we use to solve the classification and regression problems.. For both the classification and regression problem, our input consists of the k closest training examples in the feature space and the output depends on whether the k-Nearest Neighbor is used to address the classification or regression. K is a positive integer which is generally very small. k-NN being the simplest of all the machine learning algorithms, is a instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.

For k-nearest neighbors classification, the output is a class membership. We classify the object on the basis of the majority vote of its neighbors. It's throu' the votes that the object is being assigned to the class which has most common among its k nearest neighbors. If k = 1, then the object is addressed to the class of that single nearest neighbor. Whereas, for k-NN regression, the object's property value is the output. This value is nothing but the average of the values of its k nearest neighbors.

The drawback to k-NN algorithm is that it is sensitive to the local structure of the data.The algorithm is not to be confused with k-means, another popular machine learning technique.

5. Results

Confusion Matrix of the classification results on the test data using the required classifier are shown below. Rows represent the actual class and columns the predicted class. The diagonal entries show the number of test samples correctly classified.

Formula Used:
Precision Value = Correct Predictions of the Required Class / Total Predictions of the Required Class
Recall Value = Correct Predictions of the Required Class / Total Observations in the Required Class

LDA:

Activity	**Walking**	**Upstairs**	**Downstairs**	**Sitting**	**Standing**	**Laying**	**Recall %**
Walking	**490**	6	0	0	0	0	98.79
Upstairs	11	**460**	0	0	0	0	97.66
Downstairs	1	14	**405**	0	0	0	96.43
Sitting	0	1	0	**434**	56	0	88.39
Standing	0	0	0	22	**510**	0	95.86
Laying	0	0	0	0	0	**537**	100
Precision %	97.61	95.63	100	95.18	90.11	100	**96.23**

Logistic Regression:

Activity	Walking	Upstairs	Downstairs	Sitting	Standing	Laying	Recall %
Walking	**494**	0	2	0	0	0	99.6
Upstairs	23	**448**	0	0	0	0	95.12
Downstairs	4	9	**407**	0	0	0	96.9
Sitting	0	4	0	**432**	55	0	87.98
Standing	2	0	0	13	**517**	0	97.18
Laying	0	0	0	0	0	**537**	100
Precision %	94.46	97.18	99.51	97.08	90.38	100	**96.2**

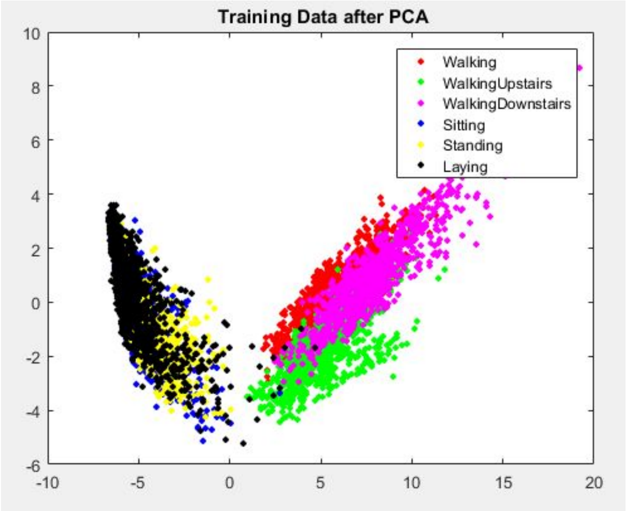
SVM for linear kernel:

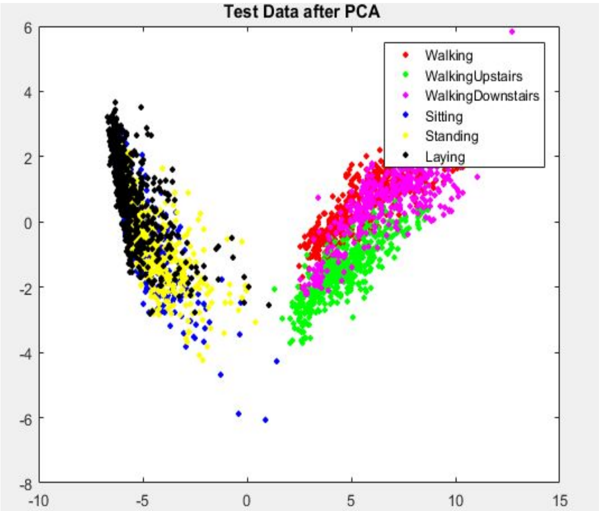
Activity	Walking	Upstairs	Downstairs	Sitting	Standing	Laying	Recall %
Walking	**492**	1	3	0	0	0	99.19
Upstairs	18	**451**	2	0	0	0	95.75
Downstairs	4	6	**410**	0	0	0	97.62
Sitting	0	2	0	**435**	54	0	88.59
Standing	0	2	0	16	**516**	0	96.99
Laying	0	0	0	0	0	**537**	100
Precision %	95.72	98.04	98.8	96.45	90.53	100	**96.4**

KNN:

Activity	Wng	Upstairs	Downstairs	Sitting	Standing	Laying	Recall %
Walking	**486**	1	9	0	0	0	97.98
Upstairs	42	**426**	3	0	0	0	90.45
Downstairs	51	42	**327**	0	0	0	77.86
Sitting	0	4	0	**420**	67	0	85.54
Standing	0	0	0	51	**481**	0	90.41
Laying	0	0	0	2	1	**534**	99.44
Precision %	83.94	90.46	96.46	88.79	87.61	100	**90.74**

We first decompose the test data to 2-dimension in order to plot the classification as shown in the given figure. We can easily figure out that the dataset is more linear separable, and we assume that SVM with linear kernel function will get the highest accuracy among other two functions.





6. Discussion

So, we have successfully applied LDA, Logistic Regression, K-Neighbors Classifier and SVM to our dataset. Now we can compare the results for all the above mentioned algorithms and can find the best fit classifier for Human Activity Recognition Dataset. The table shows the comparison of the classification results on the test data using the training related classifiers.

Classifier	Accuracy	Error Rate
SVM with linear kernel	96.4	3.60
LDA	96.23	3.77
Logistic Regression	96.2	3.80
k-NN	90.74	9.26

On the basis of accuracy, the models are ordered as SVM with kernel = ‘linear’ > LDA > Logistic Regression > k-NN.

Hence, we can say that SVM with linear kernel approach is the best fit classifier to our dataset. The SVM with linear kernel gives the best performance for all the accuracies and error rates. SVM with linear kernel being a flexible approach which is capable to reduce overfitting. And SVM with linear kernel performs astonishingly well for this dataset as linear separable data. But for other different data, it would not have such good performance.

7. Conclusion

In this report, we discussed and examined various classifiers (LDA, Logistic Regression, k-Nearest Neighbors and Support Vector Machines) and their performance in classifying human activities using smartphones, and found that their accuracies are quite different from each other. We tested different approaches and algorithms on our dataset. Overall, our list of classifiers achieved relatively high performance. While the various models displayed similar test errors, the accuracy for individual users and specific activities did vary significantly. We also concluded that on changing the kernel, Support Vector Machine classifier gives different results and amazingly find SVM with linear kernel provides a much more fine accuracy than all the other classifiers used. SVM has the lowest error rate, making it a best fit to the Human Activity Recognition data set. Whereas, k-NN has the highest error rate of 9.26 and lowest performance accuracy of 90.74, making it not the best fit for our dataset.

We found that the most of the errors are due to confusion of the activity ‘sitting’ and activity ‘standing’ labels. To classify these two activities is difficult. Therefore, we suggest that future authors should focus on creating a more effective classifier for only those classes and perhaps having additional features to distinguish sitting from standing could help in this aspect.

7. Acknowledgement

We’ve taken efforts in this project. However, it would not have been possible without the kind support of many individuals. We would like to extend our sincere thanks to all of them. We thank Professor J. Braun and Teaching Assistant Xinzi Sun for their instructions, for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. My thanks and appreciations also go to all the people who have willingly helped us out with their abilities.

References

- Bishop, C. M., ‘Pattern Recognition and Machine Learning,’ Springer, 2006
- Davies, N., Siewiorek, D.P., Sukthankar, R.: Activity-based computing. IEEE Pervasive Computing 7(2) (April 2008) 20-21
- .R. Poppe. Vision-based human motion analysis: An overview Computer vision and image understanding, 108(1):4–18, 2007.
- N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In AAAI,volume 5, pages 1541–1546, 2005
- J.-L. Reyes-Ortiz, L. Oneto, A. Ghio, A. Sama, D. Anguita, and X. Parra. Human activity recognition on smartphones with awareness of basic activities and postural transitions.In Artificial Neural Networks and Machine Learning–ICANN 2014, pages 177–184. Springer, 2014
- C.-C. Yang and Y.-L. Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. Sensors , 10(8):7772–7788, 2010.
- Bao, Ling and Stephen S. Intille. Activity recognition from user-annotated acceleration data, 2004
- http://multivariatestatsjl.readthedocs.io/en/latest/mclda.html

Contributions:

Aanchal Sharma : SVM, KNN; Anuroop Rao: Logistic Regression and Manasee Moghey: LDA, PCA