Report on Mini-project

1 Introduction

Human-centered computing is an emerging research field that aims to understand human behavior and integrate users and their social context with computer systems. The recognition of human activities has become a project with high interest within the field. Human Activity Recognition(HAR) aims to identify the actions carried out by a person given a set of observations of him/herself and the surrounding environment[1]. Recognition can be accomplished by exploiting the information retrieved from various sources such an environmental or body-worn sensors[2,3].In the past, researches have used many devices which can attach to human body to record human activities. Now, the use of smartphones with inertial sensors is an alternative solution for HAR. Compared to the body-worn sensors, smartphones have lots of advantages. It is more controllable and will have no effects on human's activity.

Activity recognition is important in many real applications. As one branch of human computer interaction, it makes the computer even "smarter", that is, it could provide the corresponding services based on what the user is doing[4]. For example, the phones detects that the users is about to leave the room, so a reminder may pop up with a message that "See the weather today", or "See the traffic situation around you ". Apart from daily life, activity recognition can also protect people from dangerous condition. For instance, when you are moving, it can access to the outer environment so to detect the obstacles on the road, this may prevent older people from falling and help people who are driving. HAR is a field of Deep Learning for its quantity of data needed and the result it wants to get.

2 Data Collection and Preparation

The data was from UCI Machine Learning Lab. The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, they captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz[5].

It this project, I have totally 561 features and 10299 rows of data. And at the start of the project, I use the function from sklearn library in python randomly split the dataset into training and testing set. The training data set contains 80 percent of the whole dataset, the testing data set has the left 20 percent.

3 Exploratory Analysis

The dataset has 10,299 records and 30 different subjects. Fig.1 shows the recorded counts for each activity.

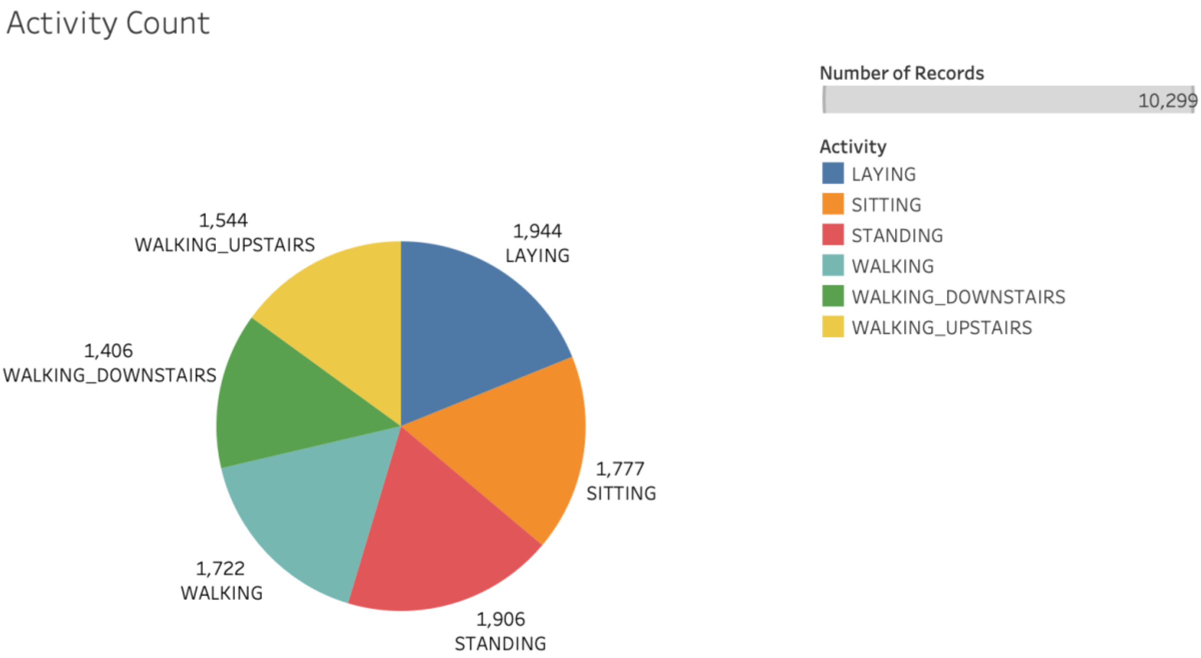


Fig.1 Activity Count

4 Methods

4.1 SVM

In this project, I choose SVM as the first predictive model. SVM(Support Vector Machine) is a linear model for classification and regression problems. The basic idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. It can separate linear or non-linear data.

For the linear SVM, we are given a training dataset of n points of the form $(x_1,y_1)\dots(x_n,y_n)$, where y_i are either 1 or -1, each indicating the class to which the point x_i belongs. We want to find the "maximum-margin hyperplane" that divides the group of points x_i for which $y_i = 1$ from the group pf points for which $y_i = -1$, which id defined so that the distance between the hyperplane and the nearest point x_i from either group is maximized. And hyperplane can be written as the set of points x satisfying $w \cdot x - b = 0$, where w is the normal vector to the hyperplane[5]. The hyperplane is like Fig.2.

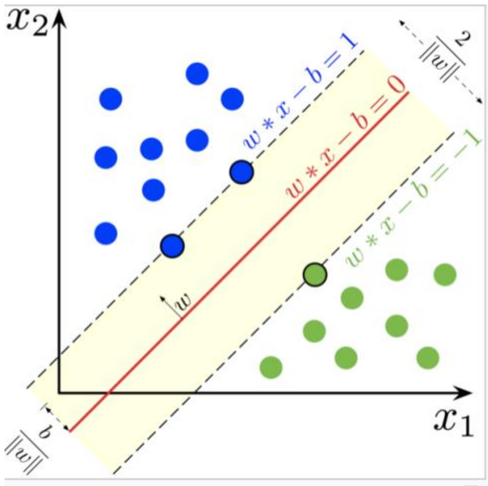


Fig.2 Linear Separable Patterns

For the non-linear separable situation, we can apply the kernel trick[6] to maximum-margin hyperplanes. The algorithm has a little different which is that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. It is like Fig.3. Since we don’t use kernel method is this project, so there is no further description in this report.

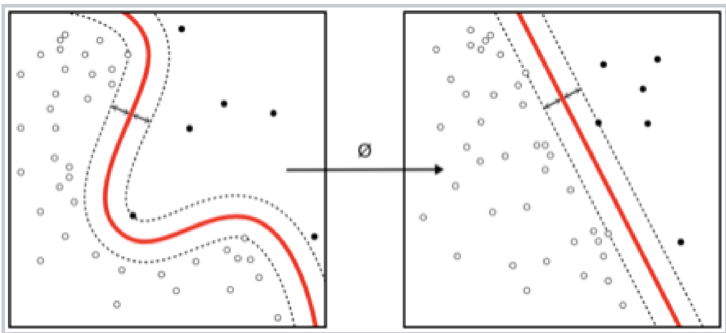


Fig.3 Kernel Function

4.2 KNN

KNN can be used for both classification and regression predictive problems. The basic idea of KNN is that if we intend to find out the class of one item, we need to evaluate the k nearest neighbors around the target, and we take vote from them. Fig.4 shows the progress of KNN. In this progress, the most important thing is that how to choose the proper k.

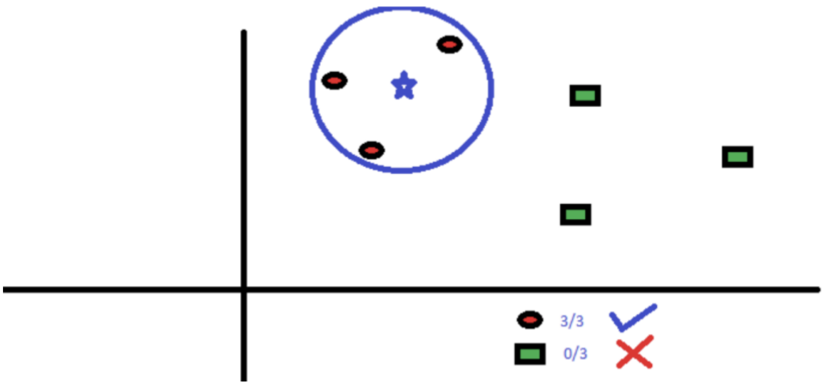


Fig.4 KNN

Typically, when k = 1, the error rate for predicting the target is zero since the nearest one neighbor is always itself. In many cases, error rate will decease with the increase of k, but it will reach a limit. For example, if we add a green point in Fig.4 and we set k equals to 7, then the blue star must be classified wrongly.

4.3 Feature Selection

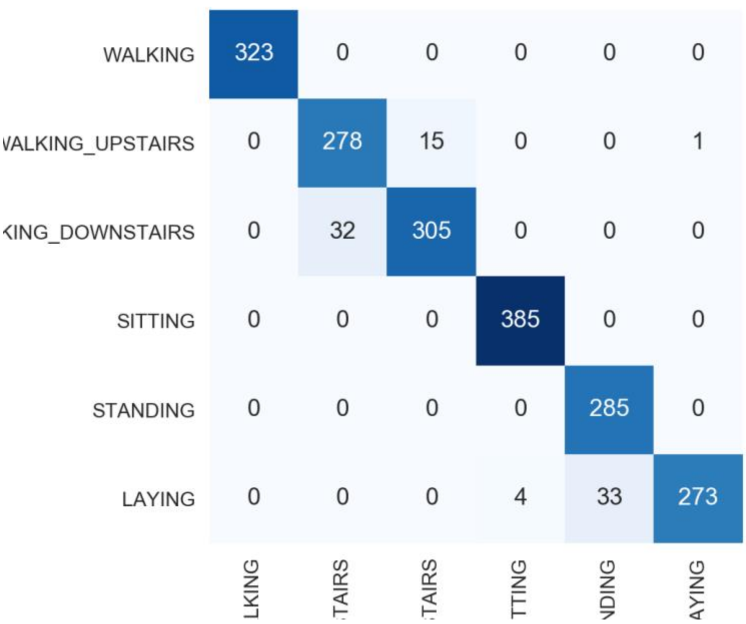
Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in [7]. Irrelevant features in your data can decrease your accuracy of the models. In this project, we get 561 features which is a huge amount of data, to use all the features will increase out training time and take up more sources. In this project, I use feature_selection function in sklearn with Python to vary the number of features. From 0 to 100 percent, every step is 5 percent more, to plot the “accuracy vs percentiles of features”.

4.4 PCA

PCA(Principal component analysis) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables(entities of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components[8]. Since the dataset we get is 3-D data so here I use PCA to do dimension reduction.

5 Results

Fig.5 shows the result of SVM model.



(a)

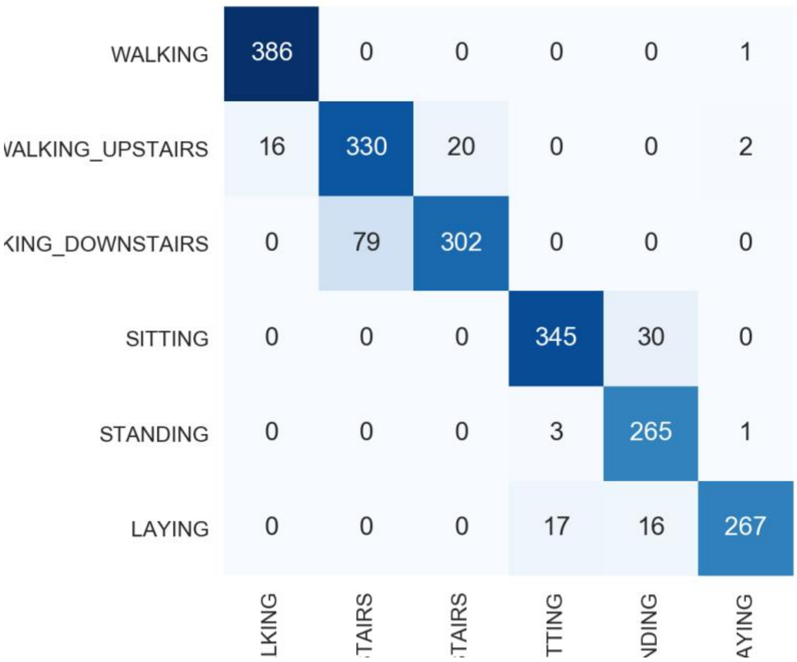
	precision	recall	f1-score	support
LAYING	0.96	1.00	0.98	387
SITTING	0.94	0.92	0.93	368
STANDING	0.97	0.94	0.95	381
WALKING	0.97	0.99	0.98	375
WALKING_DOWNSTAIRS	0.96	1.00	0.98	269
WALKING_UPSTAIRS	0.99	0.94	0.96	300
avg / total	0.96	0.96	0.96	2080

(b)

Fig.5 Confusion Matrix of SVM

The accuracy of SVM model is 0.9639423076923077, from which we can say that SVM with linear kernel has a high accuracy on this dataset.

Fig.6 shows the result of KNN model.



(a)

	precision	recall	f1-score	support
LAYING	0.96	1.00	0.98	387
SITTING	0.81	0.90	0.85	368
STANDING	0.94	0.79	0.86	381
WALKING	0.95	0.92	0.93	375
WALKING_DOWNSTAIRS	0.85	0.99	0.91	269
WALKING_UPSTAIRS	0.99	0.89	0.94	300
avg / total	0.92	0.91	0.91	2080

(b)

Fig.6 Confusion Matrix of KNN

And the accuracy of KNN model is 0.9110576923076923.
From the results we get, SVM is a more reliable more of classifiers. However, KNN is less computationally intensive than SVM. And the difference between accuracy is light, so prediction of multi-class data should be done with KNN because it is easy to implement. Besides, the algorithm that guarantees reliable detection in unpredictable situations depends upon the data. If the data points are heterogeneously distributed both should work well. If data is homogenous to look at, one might be able to classify better by putting in a kernel in to the SVM[9].

The plot of feature selection is showed in Fig.7.

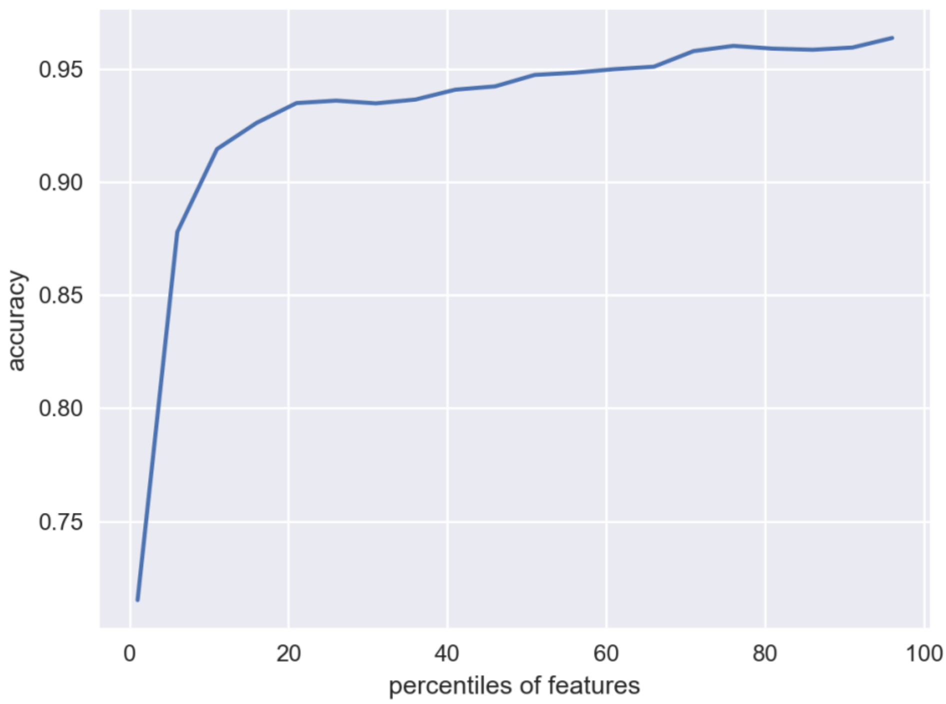


Fig.7 Accuracy vs Features

The accuracy is increasing with the increase of selected features. And when the percentage reaches to 80 percent, the curve become relatively stable. And if we want to have an accuracy above 80%, we need at least choose 28 features(5%), if we intend to get an accuracy of 90%, the number of features we selected should be more than 57(10%).

The result of PCA is showed in Fig.8.

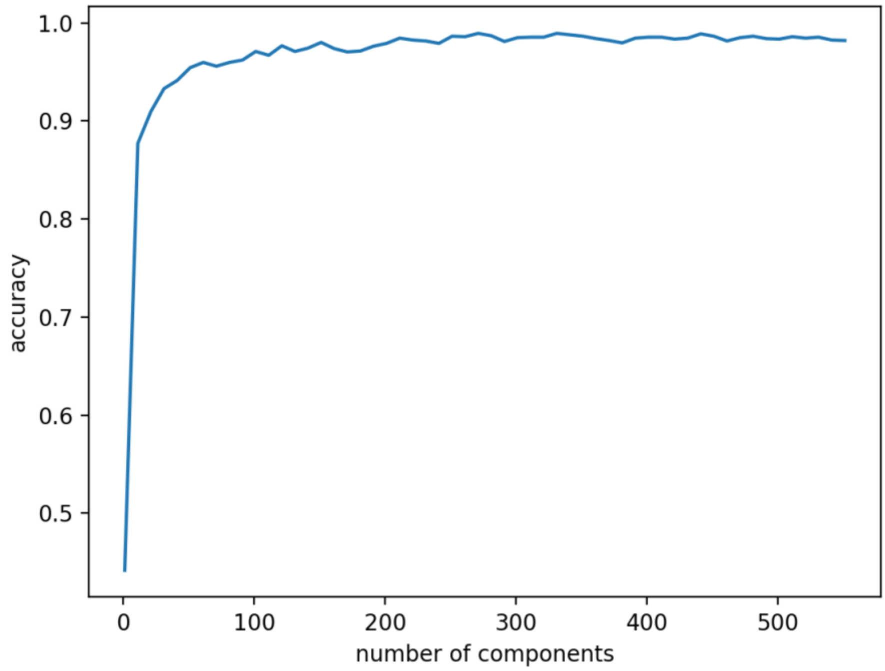


Fig.8 Accuracy vs Number of Components

The plot shows that if we want to save the training time and have a comparably high accuracy when we fit the data in SVM model, we need to set the components around 100. And If we want to have the accuracy of 90%, the components must above 18.

Reference

[1]Davide Anguita, Alessandro Ghio, Luca Oneto. A Public Domain Dataset for Human Activity Recognition Using Smartphone. Computational Intelligence and Machine Learning. Bruges(Belgium),24-26 April 2013.

[2]P.Lukowicz,J.A.Ward,H.Junker,M.Sta¨ger,G.Tro¨ster,A.Atrash,andT.Starner.Recognizingwork- shop activity using body worn microphones and accelerometers. Proceedings of the 2nd Int Conference Pervasive Computing, pages 18–22, 2004.

[3]D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine, 10(1):156–167, 2006.

[4]Xing Su, Huanghang Tong, Ping Ji. Activity Rescognition with Smartphone Sensors. Tsinghua Science and Technolog, pages 235-249, Volume 19, Number 3, June 2014.
[5]https://en.wikipedia.org/wiki/Support-vector_machine
[6]Aizerman, Mark A.; Braverman, Emmanuel M. & Rozonoer, Lev I. (1964). "Theoretical foundations of the potential function method in pattern recognition learning". Automation and Remote Control. 25: 821–837.
[7]Raheel Shaikh. Feature Selection Techniques in Machine Learning with Python. ”Towards Data Science”. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
[8] https://en.wikipedia.org/wiki/Principal_component_analysis
[9]Dr.Amita, Saarthak Mahajan. Comparison: KNN & SVM Algorithm. International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887.