PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON

Human Activity Recognition Using SVMs

Submitted by

1.	Name	Bhargavi G	SRN	PES1201800211
2.	Name	Nisha S	SRN	PES1201801777
3.	Name	Rallapalli Aiswarya	SRN	PES1201800309

Branch & Section : Computer Science Engineering, J

PROJECT EVALUATION

(For Official Use Only)

Sl.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor :

Signature of the Course Instructor :

Human Activity Recognition Using SVMs

Bhargavi Gummanur
Computer Science Engineering
PES University
Bangalore, India
bhargu2000@gmail.com

Nisha Suresh
Computer Science Engineering
PES University
Bangalore, India
nishasuresh2001@gmail.com

Aishwarya Rallapalli
Computer Science Engineering)
PES University
Bangalore, India
aiswarya$_{r}allapalli@yahoo.in$

Abstract

Human action recognition is a very important and challenging problem with various important applications in areas such as healthcare services, sports and surveillance.This project is modeled over data obtained from people performing activities of daily living by carrying a waist-mounted smartphone with embedded inertial sensors. This real-time data pre-processed into a feature vector is taken as the input. Principal Component Analysis is used to reduce the dimensions of the feature vector. Principal Component Analysis is used as it retains as much information as possible with very less information loss. It also reduces the computational complexity of the model. Another machine learning technique, Support Vector Machine(SVM) is used as a classifier. It helps in classifying the activities performed. SVMs are found to give better results when compared to other approaches. Our project confirms that our approach recognizes the human activity with a high accuracy.

Index Terms

Support Vector Machine, Principal Component Analysis, GridSearchCV

I. Introduction

Human Activity Recognition is one of the emerging fields especially in the Health Care sector. The demand for activity recognition systems have grown as these systems facilitate health workers to monitor their patients all times and detect any abnormalities.[5] These recognition systems have been accomplished in the past through motion sensors worn on different parts of the body and achieved good results with a very low error rate. But these are rather uncomfortable for the user and do not provide long term solutions. The setting is too complicated and not feasible in practice. Therefore, due to it's high computational power and commercial availability, smartphones serve as an ideal solution to this problem.[9] Smartphones are rich sources of information and can be used as sensors that identify human activities. The latest devices come with built-in sensors such as accelerometers,gyroscopes etc. that have proved worthy in activity recognition.In this project, 30 candidates were made to perform six basic activities(Walking, Walking upstairs, Walking downstairs, Sitting, Standing and Laying) wearing a smartphone on their .waists. The accelerometer and gyroscope of the device captured the 3- axial linear acceleration and the 3-axial angular velocity at a constant rate of 50 Hz. The sensor values are then preprocessed using noise filters. After filtering and transformations, we obtain a vector of 561 features obtained after calculating the mean, standard deviation, kurtosis,skewness, interquartile range,angle between the vectors etc. The dataset of these recorded values and their respective labels were available on the UCI Repository. The dataset has been divided into 70 percent training and 30 percent Testing with features as the recorded values of 516 dimension each and labels as the six activities mentioned above.

II. LITERATURE SURVEY

Research Papers and Studies gave rise to many approaches on activity recognition in the past. Some approaches include the usage of 3D Convolutional Neural Networks and SVMs [7]. The CNN approach was used to extract the spatial and temporal features from the video frames and SVMs are used to classify them into their respective activities. This algorithm was trained and evaluated on the KTH action recognition dataset. [10] Other classification methods other than SVMs have been adopted in previous works such as KNN which is an unsupervised machine learning technique that classifies the object based on the closest point that exists in the feature space. The classification depends on the labels of k nearest neighbours based on the majority vote.

III. DESIGN AND IMPLEMENTATION

A. Data Preprocessing

The data files provided are split into training data and testing data with around 7500 and 3500 rows respectively. The features provided in the dataset are in the form of string values and each value must be converted to a list of float numbers and the labels are encoded to numerical values in order to pass it to the model. The extra labels provided in the dataset are deleted as the number of datapoints for these are very less compared to the others and are not found to be biased. The pie chart for label values can be visualized.

B. Support Vector Machines

Support Vector Machine is a supervised learning model. It means you need a dataset which has been labeled. In this project, candidates were made to perform six basic activities(Walking, Walking upstairs, Walking downstairs, Sitting, Standing and Laying). These activities are the labels. SVM is then trained with the labeled data. SVM learns a linear model. It will learn a line which will be able to separate the data. This line is called a hyperplane because the support vector machine can work with any number of dimensions. An hyperplane is a generalization of a plane. In one dimension, an hyperplane is called a point, in two dimensions, it is a line,in three dimensions, it is a plane and in more dimensions we can call it an hyperplane.Fig.[1] shows, a blue line(hyperplane) linearly separates the data point in two components. This approach is selected because it can perform well even if the training data is small or has a high dimensional space [7].

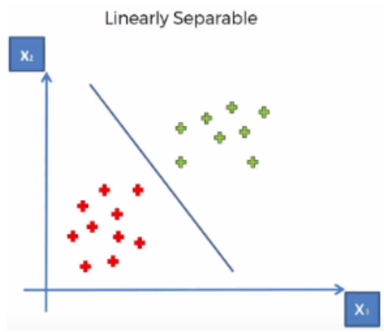

Fig. [1] Linearly Separable data

The main objective of applying SVMs is to find the best line in two dimensions or the best hyperplane in more than two dimensions in order to help us separate our space into classes [6]. The hyperplane (line) is found through the maximum margin, which is the maximum distance between data points of two classes. Maximizing the margin is the same thing as minimizing the norm of w.
For a given training data D, a set of n points of the form,

$$D = \{(x_i, c_i)|x_i \in R^p, c_i \in \{-1, 1\}\}_{i=1}^{n}$$

$$where\ c_i\ is\ either\ 1\ or\ -1,\ indicating\ the\ class\ to\ which\ the\ point\ x_i\ belongs\ [e].Each\ x_i\ is\ a\ p\ dimensional$$

vector.Any hyperplane can be written as the set of points x satisfying w.x-b=0
Where . denotes the dot product. The vector w is a normal vector, perpendicular to the hyperplane. The parameter

$$\frac{b}{||w||}$$

determines the offset of the hyperplane from the origin along the normal vector w[f]. If the training data are linearly separable, we can select the two hyperplanes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyperplanes is

$$\frac{2}{||w||}$$

, so we want to minimize w. As we also have to prevent data points falling into the margin, we add the following constraint: for each i, either

$$w.x_i - b \geq 1$$

for xi of the first class or

$$w.x_i - b \leq 1$$

for xi of the second class
This can be rewritten as:

$$c_i(w.x_i - b) \geq 1, forall 1 \leq i \leq n.$$

We can put this together to get the optimization problem (g): Minimize in (w , b) w subject to

$$c_i(w.x_i - b) \geq 1, for\ any\ i = 1, 2, ..., n(g)$$

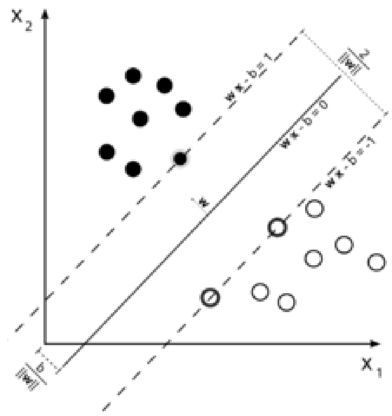

Fig. 2. Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

The above optimization is for linearly separable data. Raw data are not always linear. We come across data which are not linearly separable most of the time in our life. Fig. [3] shows non linearly separable data. We can add one extra dimension to the data points to make it separable. Fig. [4] shows linearly separable data. One dimensional data is transformed to a higher dimensional data to make it separable. The process of making non-linearly separable data point to linearly separable data point is known as Kernel Trick.

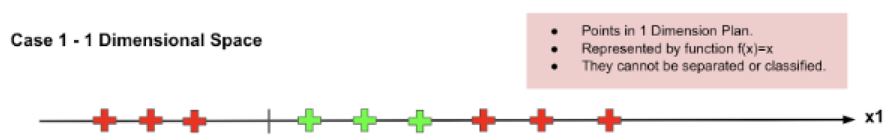

Fig. 3. Non Linearly Separable Data

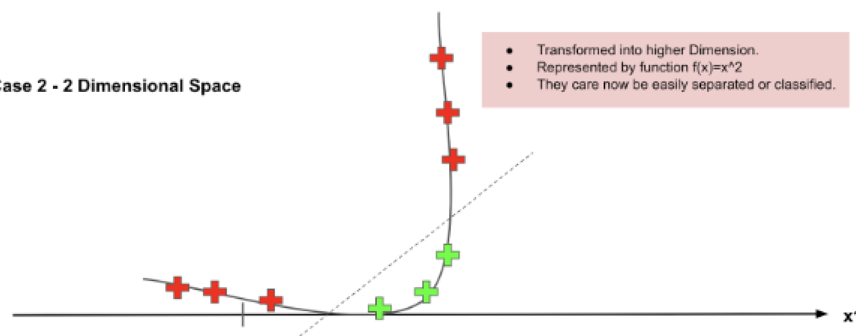

Fig. 4. Non Linearly Separable data converted to Linearly Separable data

C. *GridSearchCV*

A hyperparameter is a characteristic of a model whose value cannot be estimated from the data.This value is set before the learning process begins.[3] Grid Search is used to find the optimal hyperparameter which results in the most accurate prediction. There are three major parameters that can be tuned for Grid Search in SVMs . Kernel- The main function of the kernel is to take a low dimensional input space to a higher dimensional input space [4]. The kernel used is the Gaussian Rbf Kernel.

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$

C (Regularisation)- This parameter represents the error term which tells the SVM how much of the error is bearable. Gamma- It defines how far the distance influences the line of plausible separation.

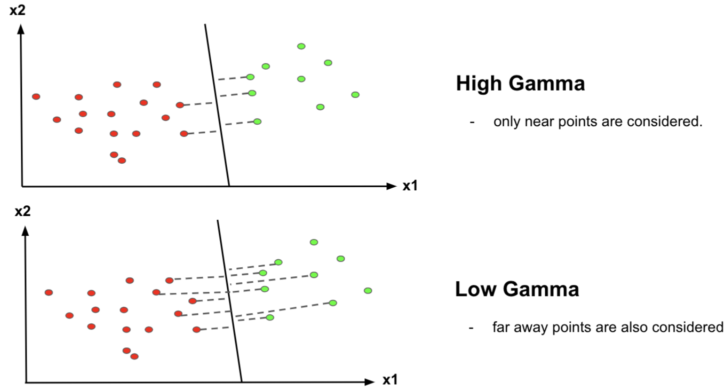

Fig. 5 Gamma

The cross validation value in the project is 8 which means that the data is divided into 8 sets and the algorithm is trained and tested 8 times. This reduces the problem of complete different accuracies on two different test sets of the same algorithm.

D. Principal Component Analysis

After preprocessing the data , we perform Principal Component Analysis on the features values to reduce the dimensions from 516 to 200. The idea of PCA is to reduce the number of dimensions of the dataset correlated to each other while preserving the variations in the dataset to a maximum extent. In order to use PCA , the values must be normalized or scaled. In short PCA is nothing but an eigenvalue method to reduce while preserving important information. The principal components are nothing but the measures of variations in the data. Basically,Principal Components are given by an orthogonal linear transformation of a set of variables optimizing a certain algebraic criterion. In our data, we reduce it to 200 dimensions, therefore , we have 200 principal components with their variations in the dataset ranging from maximum to minimum. Higher the magnitude of the PCA value higher is the influence or correlation of that data point . The two main steps required in finding out the principal components are to subtract the mean from the data point and to find the covariance matrix i.e the relation between the dimensions. The Variance of the data is defined as the deviation of each and every term from it's arithmetic mean. Co-variance is nothing but the variance that is taken with respect to multiple dimensions.

$$\mathrm{Cov}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - X')(Y_i - Y')}{(n-1)}$$

Where X' is the arithmetic mean of data X Y' is the arithmetic mean of data Y and n is the number of observations On multiplication of any random vector to this matrix, we get a vector of greater magnitude with direction turned towards the vector with highest variation that is principal component 1.[o]Principal Components are invariant under orthogonal transformations of the matrix but cannot be be applied generally under other transformations. The Principal Components are highly sensitive to the units of measurements which is also a major drawback in PCA on covariance matrices. Further multiplication with random vectors reaches a point where the direction of the vector does not change. This vector is known as an eigenvector. An Eigenvector of a matrix A is a vector when multiplied by A returns a vector which is a scalar multiple of itself [8].

EigenVectors and Eigenvalues:

Eigen vectors arise from the nature of transformation. If a transformation matrix is multiplied with a vector on the left resulting in vectors on the line y = x, this vector is referred to as an eigenvector of that transformation matrix as any vector that lies on the line y=x, it's reflection is itself. No matter how many dimensions are there, eigen vectors are always perpendicular to each other. The length or the magnitude of the vector does not affect the properties of eigen vectors but the change in direction does. The multiple of an eigenvector when transformed is the eigenvalue . The eigenvectors and the eigenvalues always come in pairs.

$$Av = \lambda v$$

To find the eigenvector, we first find the eigenvalue lambda using,

$$\det(\Sigma - \lambda I) = 0$$

Orthogonal or Orthonormal Basis of a Vector:
An orthogonal basis of a vector space V is the set of basis vectors which are mutually orthogonal . In mathematical form, two vectors are said to be orthogonal if their inner product is zero.

$$a^T b = \sum_{i=1}^{n} a_i b_i = 0 \Rightarrow \text{a perpendicular to b}$$

The inverse of an orthogonal matrix gives the transpose of this matrix. It has been proved that the columns of an orthonormal matrix are orthonormal to each other. i.e

$$(A^T A)_{ij} = a_i^T a_j$$

A matrix is symmetric if it is orthogonally diagonalizable which means that there exists E such that

$$A = EDE_T$$

where A is a symmetric matrix. It has been proved that orthogonal matrices are linearly diagonalizable if and only if the matrix eigenvectors are linearly independent. Furthermore, let E = [e1 e2 . . . en] be the matrix of eigenvectors placed in the columns. Let D be a diagonal matrix where the ith eigenvalue is placed in the iith position.

$$AE = [Ae_1, Ae_2, ...Ae_i]$$

$$ED = [\lambda_1 e_1, \lambda_2 e_2 ... \lambda_n e_n]$$

$$AE = ED- > Ae_i = \lambda_i e_i$$

With a little rearrangement, we get the equation mentioned above for orthogonal diagonalization. It can also be proved that the eigenvectors of the symmetric matrix A are all orthonormal provided they are normalized which means that E is an orthonormal matrix where

$$E^T = E^{-1}$$

PCA aims at maximizing the equations to find the maximum variance that is through finding the corresponding vectors from a covariance matrix. Covariance matrices are symmetric and symmetric matrices have orthogonal vectors. Therefore PCA always leads to orthogonal components. The two values of lambda are multiplied with the matrix to get two eigenvectors. These vectors are the principal components. Next, we project these eigenvectors onto our new dimensions, that is the reduced dimensions. The first step involved in this is to center our data around the mean i.e to subtract the mean from each datapoint. The first 200 eigenvectors are chosen and the dot product of these vectors with the transposed center data gives the projection.

$$\begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_m' \end{bmatrix} = \begin{bmatrix} (\vec{x}-\vec{\mu})^T \vec{e}_1 \\ (\vec{x}-\vec{\mu})^T \vec{e}_2 \\ \vdots \\ (\vec{x}-\vec{\mu})^T \vec{e}_m \end{bmatrix} = \begin{bmatrix} (x_1-\mu_1)e_{1,1} + (x_2-\mu_2)e_{1,2} + \cdots + (x_d-\mu_d)e_{1,d} \\ (x_1-\mu_1)e_{2,1} + (x_2-\mu_2)e_{2,2} + \cdots + (x_d-\mu_d)e_{2,d} \\ \vdots \\ (x_1-\mu_1)e_{m,1} + (x_2-\mu_2)e_{m,2} + \cdots + (x_d-\mu_d)e_{m,d} \end{bmatrix}$$

Where d is the original dimension and m is the required dimension. Hence, the resultant vector comprises the coordinates in all 200 dimensions. This is the PCAd vector which is the representation of the original features of 516 dimension.

IV. EVALUATION METRICS

The Gamma Value of 0.001, the rbf kernel and the best C value of 100 are the outputs given by the GridSearchCV . We obtain a training score of 0.9875927886934094 and a testing score which is also the accuracy of the model of 0.954272.

	True	Predicted
0	3	3
1	5	5
2	4	4
3	4	4
4	1	1

Fig. 6 represents the confusion matrix between the predicted and the true values. Most of the labels have been predicted correctly. The color of the matrix also depends on the number of data points.

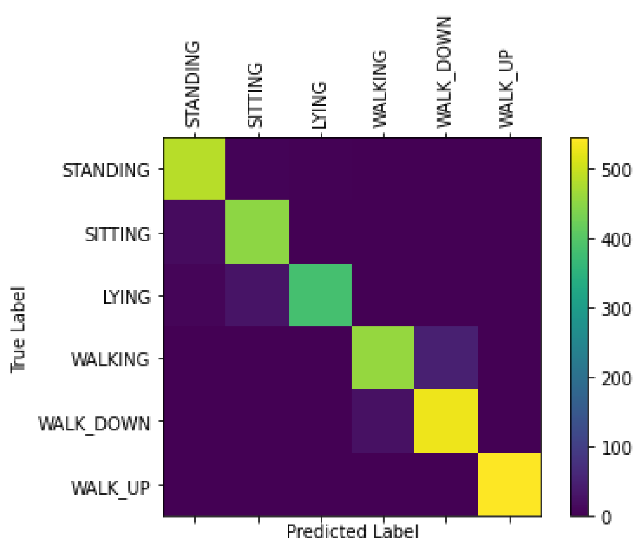

Fig. 6 Confusion Matrix

```
[[486   6   4   0   0   0]
 [ 16 453   2   0   0   0]
 [  7  27 386   0   0   0]
 [  0   1   0 458  48   1]
 [  0   0   0  22 534   0]
 [  0   0   0   0   0 545]]
```

Fig. 7. Confusion Matrix

To evaluate our performance we observe the precision and the recall values where the precision is the ratio of the correctly predicted positive values to the total number of positive values [2].

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP}$$

Where TP is True Positive and FP is False Positive.
The Recall value is the ratio of correctly predicted positive to all the observations in the class. If the recall value of the model is above 0.5, the model is said to perform well.

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

The F1- score is the weighted average of the precision and the recall therefore taking both false positives and false negatives into account. The F1 score works better when there is an uneven class distribution. If the cost of false positives and false negatives are very different, we take the F1 score into consideration or else the accuracy works perfectly fine.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$TP + TN$$

	precision	recall	f1-score	support
0	0.9798	0.9529	0.9662	510
1	0.9618	0.9283	0.9447	488
2	0.9143	0.9846	0.9481	390
3	0.8996	0.9481	0.9232	482
4	0.9550	0.9155	0.9349	580
5	1.0000	0.9982	0.9991	546
accuracy			0.9533	2996
macro avg	0.9518	0.9546	0.9527	2996
weighted avg	0.9543	0.9533	0.9534	2996

Fig. 8 Classification

V. DISCUSSION

Our model classifies activity Walking Upstairs without any misclassification (Fig. 8). In 4th row, 5th column in Fig. 7 we have a value of 48 which indicates that 48 readings out of 508 of the class walking is misclassified as walking down. 27 values of class Lying have been misclassified as Sitting. The most confusion is for class walking and the best results are achieved with walking upstairs class. The model gives an accuracy of 95.33%. We have referred to Human action recognition using support vector machines and 3D convolutional neural networks paper [7]. 3D Convolutional Neural Networks

(CNNs) were used as a feature extractor method based on spatial and temporal dimensions. Extracted features were classified by support vector machines algorithm. Their proposed system which was trained and evaluated on the KTH dataset gave an accuracy of 90.34% We can say that our model has a better performance compared to the paper we referred to [7].

VI. CONCLUSION

In this paper, we have presented Support Vector Machine(SVM) and Principal Component Analysis(PCA) models for classifying human activities. Principal Component Analysis is used for the dimensionality reduction of feature vector and Support Vector Machine is used for classification based on data obtained from PCA. The confusion matrix obtained confirms that our model recognises the activities performed with high accuracy.Our research project as of now is restricted to the basic six activities mentioned above.This work can be optimized and extended to detect other activities like sneezing or coughing.The same idea can also be used to detect abnormal behaviour or activities which can prove to be useful in many applications such as healthcare or crime detection.

VII. REFERENCES

[1] Deng, N. Tian, Y. Zhang, C..(2013). Support vector machines: Optimization based theory, algorithms, and extensions, CRC Press, Taylor Francis Group, 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742, (2:45-47).

[2] https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

[3] https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e

[4] https://towardsdatascience.com/svm-hyper-parameter-tuning-using-gridsearchcv-49c0bc55ce29

[5] https://www.hindawi.com/journals/jece/2015/140820/

[6] https://www.kdnuggets.com/2020/03/machine-learning-algorithm-svm-explained.html

[7] Latah, Majd. (2017). Human action recognition using support vector machines and 3D convolutional neural networks. International Journal of Advances in Intelligent Informatics. 3. 10.26555/ijain.v3i1.89.

[8] Mishra, Sidharth Sarkar, Uttam Taraphder, Subhash Datta, Sanjoy Swain, Devi Saikhom, Reshma Panda, Sasmita Laishram, Menalsh. (2017). Principal Component Analysis. International Journal of Livestock Research. 1. 10.5455/ijlr.20170415115235.

[9] Rasekh, Amin Chen, Chien-An Lu, Yan. (2014). Human Activity Recognition using Smartphone.

[10] Zahraa Al-asady and Amel Al-amery 2019 IOP Conf. Ser.: Mater. Sci. Eng. 518 052008 https://iopscience.iop.org/article/10.1088/1757-899X/518/5/052008/pdf

[11] Jolliffe, I.T. 1986. Principal Component Analysis. Springer-Verlag, Berlin. (2ed., Springer, 2002) Principal Components Using a Correlation Matrix 2:21