Human Activity Recognition

Yifan Zhang
ivana.zhang@mail.utoronto.ca

Abstract

Human Activity Recognition (HAR) aims to recognize the human actions from observations on movement and the environmental conditions. The purpose of this project is to recognize human activities while using mobile sensor data. In this paper, we are going to use some unsupervised methods in visualization of data. After using PCA as feature reduction to improve the efficiency, we test different machine learning approaches to predict six different human activities (Lying, Sitting, Standing, Walking, Walking Downstairs and Walking Upstairs).

1 Introduction

Smarting environment and applications are the inevitable trend with people's intrinsic needs of specialized services in terms of healthcare, safety and home automation. In order to improve our understanding of the human activities, we need to collect action information of human beings and analysis it efficiently. The rapid growth of technology provide a wide range of sensors and portable computing devices, which could be used to record information from human's daily life. Wearable devices with inertial sensors are able to provide a great variety of information of activities indoors or outdoors. The most used HAR sensors are accelerometers, but may also include gyroscopes, magnetometers, altimeters, microphones, and video cameras [2].

Since smartphones are an integral part of contemporary life, whose built-in sensors can provide abundant of effective information for HAR and be useful to a variety of applications. For example, we could keep track of bio-signal of users which is helpful to access health status of users. Also, some smartphone applications use data to estimate basic fitness statistics, such as daily step count.

In this paper, the dataset I propose to use is the Human Activity Recognition Using Smartphones Data Set [1], which recorded 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors(accelerometer and Gyroscope). At first, we use unsupervised method t-SNE, PCA, Kernel PCA to visualize dataset in 2-dimensions which could help us understanding the dataset. After analyzing, we use PCA to reduce the observations to 150 dimensions and compare the performance of different classifiers: Logistic Regression, SVM, DT and K Nearest Neighbor. The overall idea of the model is shown in Figure 1.

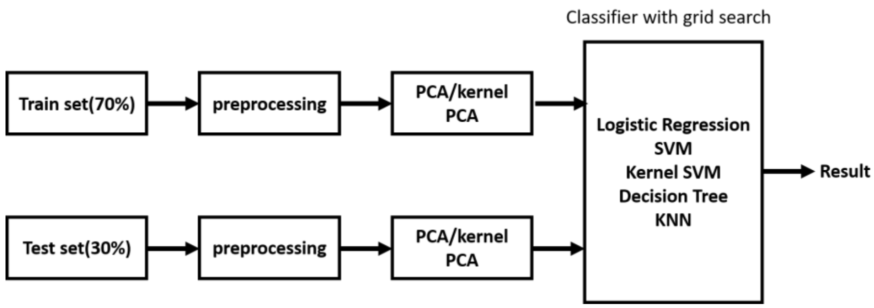

Figure 1: Overview of model

2 Related work

Human Activity Recognition (HAR) is a research field that aims to identify the actions carried out by one or more subjects through the gathering and understanding of context information about the user state and its surrounding environment [3]. HAR systems employ three main subsystems: data collection, model parameters, activity recognition [2].

There has been a variety of work done in this field. The earlier work investigated the possibilities to use accelerometers as main sensor data for activity recognition [3]. The most common machine learning algorithm of activity recognition is SVM [3]-[5], while Decision Tree (DT), Bayesian Network (BN), Naïve Bayes (NB) and K Nearest Neighbor (KNN) could also be used as classifiers [5].

Unlike the methods proposed by other researches, in this project, we try to analysis the dataset by Principal Component Analysis (PCA) and t-SNE at first, then we use PCA to reduce the dimensions and apply different machine learning algorithms on prepared data.

2 Formal description

2.1 Dataset

In this project, we use the public available dataset from UCI Machine Learning Repository [1], which is built to recognize the movements of human by the inertial sensors of smartphones.

A group of 30 volunteers with ages ranging from 19 to 48 years were selected for this task. Each person was instructed to follow a protocol of activities while wearing a waist-mounted Samsung Galaxy S II smartphone. The six selected ADL were standing, sitting, laying down, walking, walking downstairs and upstairs [2].

By using the sensors (Gyroscope and accelerometer) in a smartphone, they have captured '3-axial linear acceleration'(tAcc-XYZ) from accelerometer and '3-axial angular velocity' (tGyro-XYZ) from Gyroscope with several variations. These sensor signals are preprocessed by applying noise filters and then sampled in fixed-width windows (sliding windows) of 2.56 seconds each with 50% overlap. There are total 564 features of each observation [2]. Separating dataset into training set (70%) and test set (30%).

2.2 Analysis by unsupervised methods

2.2.1 Principal Component Analysis (PCA) and Kernel PCA

Principal Component Analysis is being used to identify the directions in feature space also called as "Principal Components" along which data varies the most [6].

Let $\mathbf{x}=[x_1,x_2,...x_n]^T$ be an n-dimensional random vector having no mean. The matrix V defines the principal direction of the projection, which could let reduce k-dimensional projection $\mathbf{x}' = \mathbf{V}\mathbf{x}$ remains as much of the variance of X as possible. We could compute by eigendecomposition $\mathbf{C} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T$ of the sample covariance matrix $\mathbf{C} = E(\mathbf{x}\mathbf{x}^T)$. The eigenvalues $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ determine the $\mathbf{x}' = [x'_1, x'_2, ..., x'_n]^T$ are computed by projection the original data to the principal directions $\mathbf{x}' = \mathbf{E}^T\mathbf{x}$

We perform PCA and Kernel PCA (rbf) in two dimensions in Figure 2. We could see that PCA and kernel PCA could separate dynamic action (Walking, Walking Upstairs and Walking Downstairs) and static actions (Standing, Sitting and Laying) well.

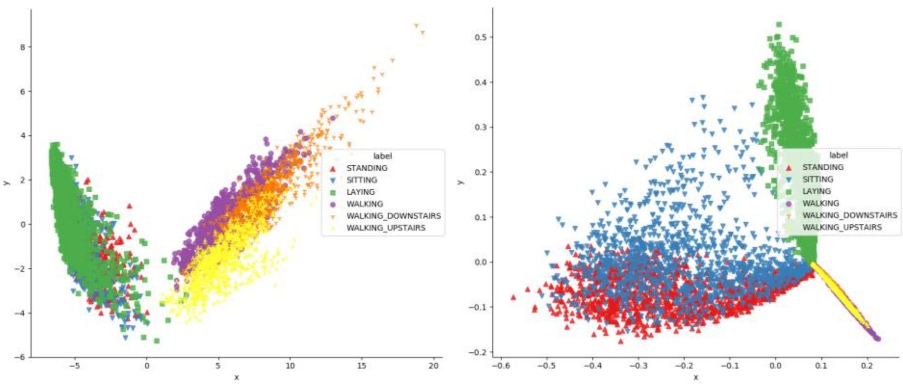

Figure 2: first two components of PCA and Kernel PCA, red: Standing, blue Sitting, green Laying, purple: Walking, orange: Waking Downstairs, yellow, Walking Upstairs.

2.2.2 t-SNE

t-SNE is a technique for the visualization of similarity data is capable of retaining local structure of the data while also revealing some important global structure of the data (such as clusters at multiple scales) [7]. We use t-SNE to perform the dataset in Figure 3.

From the figure 3 we could see that sitting and standing is completely overlay, while laying is well clustered together. Walking Upstairs and Walking Downstairs is clustered except few data points while Walking is distributed randomly.

.

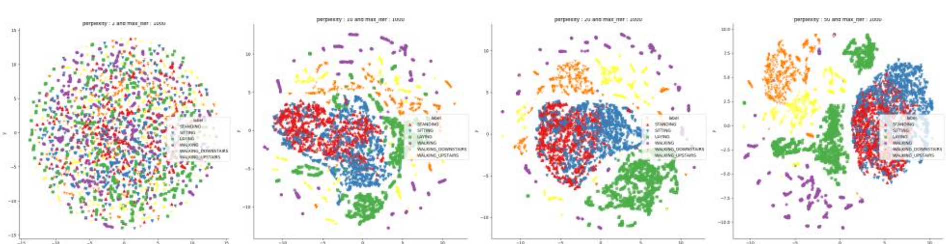

Figure 3: perform t-SNE with perplexity = [2, 10, 20, 50], red: Standing, blue: Sitting, green: Laying, purple: Walking, orange: Waking Downstairs, yellow, Walking Upstairs.

2.3 Dimensionality reduction

Feature reduction techniques attempt to reduce dimensionality by discarding some of the original features, whereas feature transform methods attempt to map the original features into a lower dimensional subspace [4]. In this project, we use PCA to reduce the features and expected that they would reveal essential information in acceleration signals that describe human activity. We compare PCA and Kernel PCA in Figure 4. It could be seen that after taking 150 components of PCA, the percentage could reach 99 percent. It is more likely that the components above 150 will have very less impact on the classification.

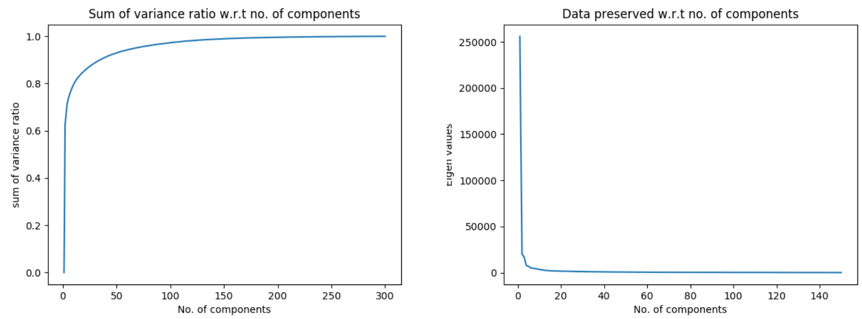

Figure 4: the figure on the left show the percentage of variance explained by the selected components when applying PCA. The figure on the right show eigenvalues of the centered kernel matrix in decreasing order.

We use Logistic Regression as example to compare the appearance of PCA and Kernel PCA in Figure 5. It could be seen that PCA always has better performance than Kernel PCA. So in this project, we choose to use PCA and the components is 150.

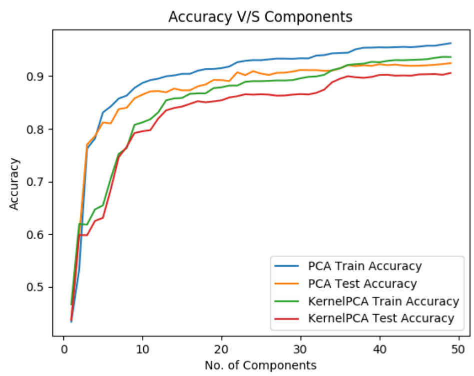

Figure 5: the train and test accuracy of Logistic Regression explained by the selected components when applying PCA and Kernel PCA.

3 Modal Comparison

After reducing dimensionality to 150 by PCA on 30 users performing six different activities, we try to find the best classification model among Logistic Regression, K-Nearest Neighbors (KNN), Support vector machine (SVM), Kernel Support vector machine (SVM) and Decision Tree. The above classifiers are available in the scikit-learn library. We also use grid search to find the proper parameters of each algorithms.

3.1 Logistic Regression

Logistic Regression is a model which really fast at classification and always has good accuracy, so we choose it at first. We try two different penalty ‘l1’ and ‘l2’ and inverse of regularization strength C for 0.01, 0.1, 1, 10, 20 and 30. The best parameters we finally choose is ‘l2’ and 20, which average cross validate scores of best estimator is 0.94015. The confusion matrix of test set is shown in Figure 5.

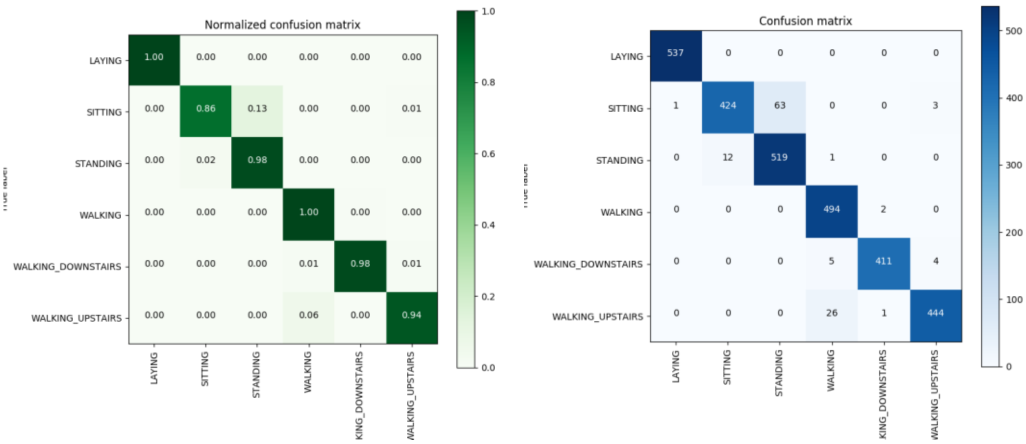

Figure 5: confusion matrix of test set

3.2 Support vector machine

The SVM algorithm predicts by computing the hyperplane with largest margin as the classification separation plane. Just like Logistic Regression, we try penalty ‘l1’ and ‘l2’ and penalty parameter C for 0.125, 0.5, 1, 2, 8 and 16. The average 3 fold cross validate scores of best estimator is 0.9412, when C is 0.5 and the penalty function is l2. The confusion matrix is shown in Figure 6. We also try SVM with kernel rbf, which achieves highest accuracy 0.9531 with the best parameters C=16, gemma=0.0078125.

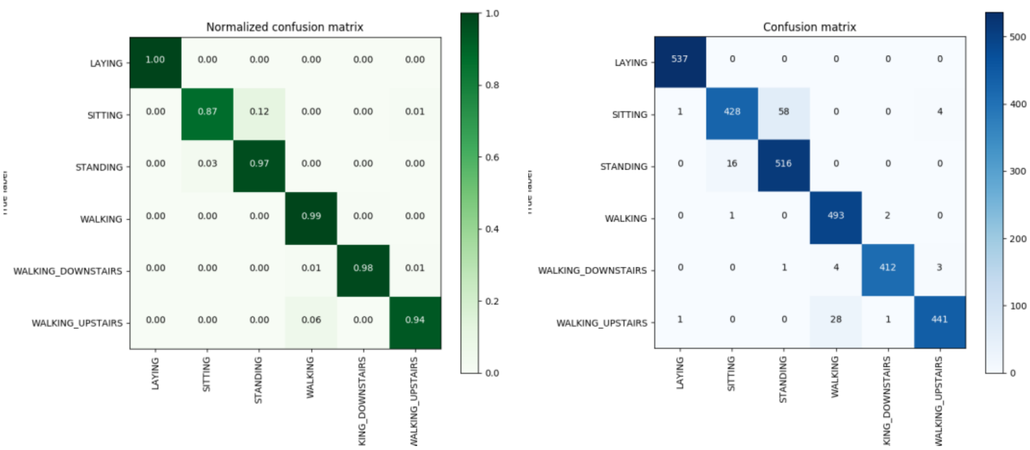

Figure 6: confusion matrix of test set

3.3 Decision Tree

Decision Tree predicts using a tree-like graph of decisions. However, Decision Tree performs really badly in our experiments. We try to improve the performance by tune the hyper parameter max_depth. But the average 3 fold cross validate scores of best estimator is only 0.8019(max_depth=7), which is still worse than expectation. The confusion matrix is shown in Figure 7. It could be seen that DT is really bad at distinguishing Sitting and Standing. It also show a unexpected result on separate static actions(Walking, Walking Upstairs and Walking Downstairs)

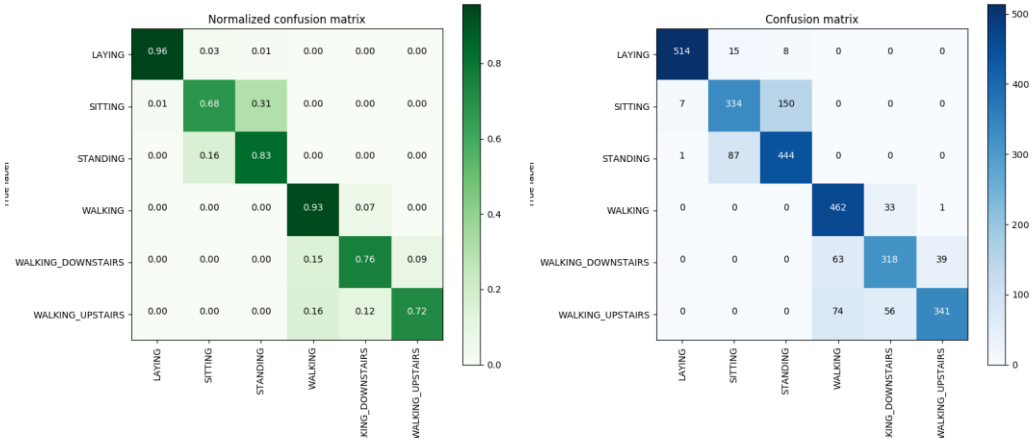

Figure 7: confusion matrix of test set

3.4 KNN

The KNN algorithm predicts according to the nearest neighbors in feature space. We compare kNN models with different k numbers. The best parameters get is 9. However, because of the large variance between different sensors, we do not get a good result from this method, the average 3 fold cross validate scores of best estimator is 0.8917. The confusion matrix of d test set is shown in Figure 8.

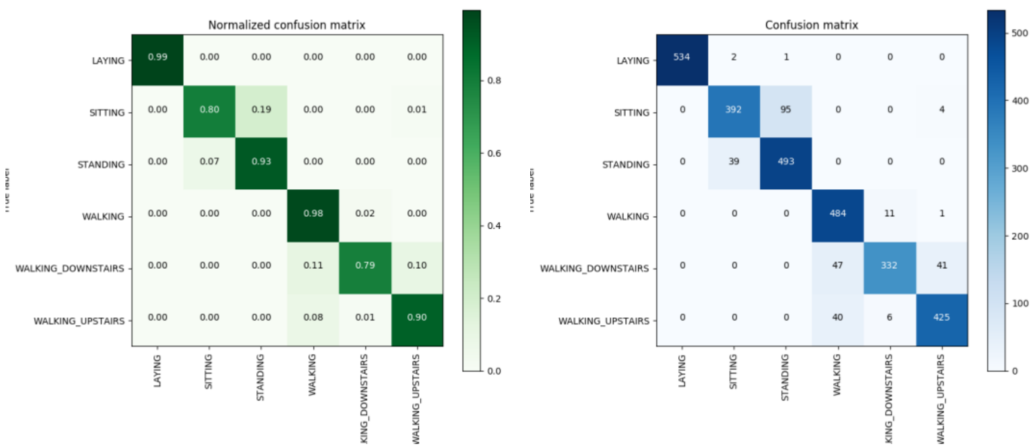

Figure 8: confusion matrix of test set

3.5 Discussion

Based on the data we got from experiments, we could summarize the result in Table 1:

Table 1: Model comparison

Classifier	Running time	Accuracy	Best score
Logistic Regression	2.6min	0.9600	0.9401
SVM	30.2s	0.9593	0.9412
Kernel SVM(rbf)	2.4min	0.9602	0.9531
Decision Tree	11.2s	0.8188	0.8019
KNN	50.1s	0.9026	0.8917

On the basis of accuracy, the model are ordered as Kernel SVM > SVM> Logistic regression > KNN > Decision Tree. It's clearly that Kernel SVM is the best classifier for this model. The classification report of Kernel SVM is shown in Table 2. It could be seen that Sitting and Standing is the action with highest error rate, which is the part our method needs to be improved.

Table 2: classification report of Kernel SVM

	precision	**recall**	**f1-score**	**support**
Laying	1.00	1.00	1.00	537
Sitting	0.96	0.88	0.92	491
Standing	0.90	0.96	0.93	532
Walking	0.96	0.98	0.97	496
Walking Downstairs	0.98	0.94	0.96	420
Walking Upstairs	0.94	0.96	0.95	471
Average	0.96	0.96	0.96	2947

5 Limitation

1. Sitting and standing is the most difficult to separate for each classifier. Even for the best classifier Kernel SVM, we still got f1-score 0.93, 0.92 for sitting and standing respectively. We think it is because these activities are similar in the view of sensors. Maybe adding more data could help to distinguish those two actions. Or we could try to find the feature that could separate Sitting and Standing well and add it to our feature selection.

2. Decision Tree shows a poor performance in our experiments. The best score of DT could only achieve 0.8. In our opinion, this might due to the complexity of the data features. Maybe we could try to ensemble Decision Tree with bagging and boosted method and compare the performance in the future work.

6 Conclusion

This project presents experiments for smartphone-based human activity recognition by using Principal Components Analysis and classification algorithms. The selected feature set have remarkable time efficiency along with comparable classification accuracy up to 96%, which shows that our method is possible to identify the behaviors of users with only 150 features of observations.

References

[1].UCI Machine Learning Repository, Activity Recognition Using Smartphones Data Set https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[2].Tundo, M. D. (2014). Development of a human activity recognition system using inertial measurement unit sensors on a smartphone (Doctoral dissertation, Université d'Ottawa/University of Ottawa).

[3].Anguita, D., Ghio, A., Oneto, L., Llanas Parra, F. X., & Reyes Ortiz, J. L. (2013). Energy efficient smartphone-based activity recognition using fixed-point arithmetic. Journal of universal computer science, 19(9), 1295-1314.

[4].He, Z., & Jin, L. (2009, October). Activity recognition from acceleration data based on discrete consine transform and SVM. In Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on (pp. 5041-5044). IEEE.

[5].Lau, S. L., & David, K. (2010, June). Movement recognition using the accelerometer in smartphones. In Future Network and Mobile Summit, 2010 (pp. 1-9). IEEE.

[6] Schölkopf, B., Smola, A., & Müller, K. R. (1997, October). Kernel principal component analysis. In International Conference on Artificial Neural Networks (pp. 583-588). Springer, Berlin, Heidelberg.

[7] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9.