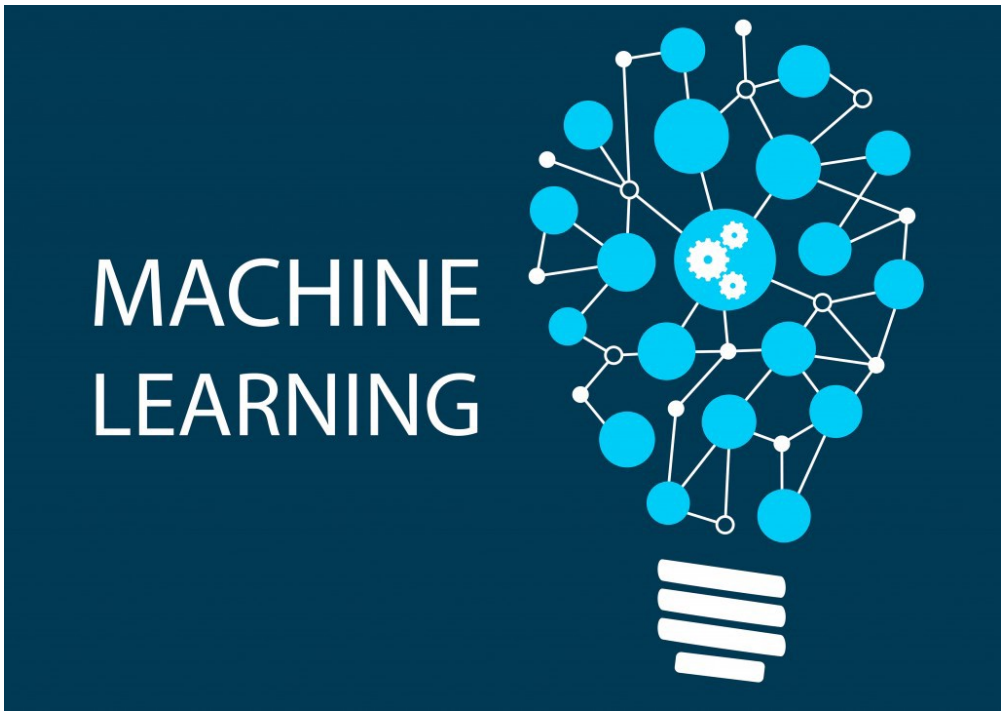
Project Report



Analysis of Human Activity Recognition Using smartphones

CISC 5800 Machine Learning

Wanjia Song

Table of Contents:

I. Introduction

For my project, I choose the dataset of Smartphone-Based Recognition of Human Activities and Postural Transitions Dataset. The goal of this project is to predict a person's human activities (eg. Walking, Walking_upstairs) through the dataset accelerometer and gyroscope 3-axial raw signals with 12 different types of activity status. To achieve this goal, I focus on four parts of the analysis 1) A general review of training dataset and test dataset 2) A Principle Component Analysis to the dataset 3) Two different classification methods to predict the labels 4) A summary for this report

II. General review of the dataset

The dataset used in my project is from accelerometer and gyroscope 3-axial raw signals. The training set has 7767 records and 561 features with 12 labels. The test dataset has 3162 recodes and 561 features also with 12 labels. All of the values are between -1 and 1.

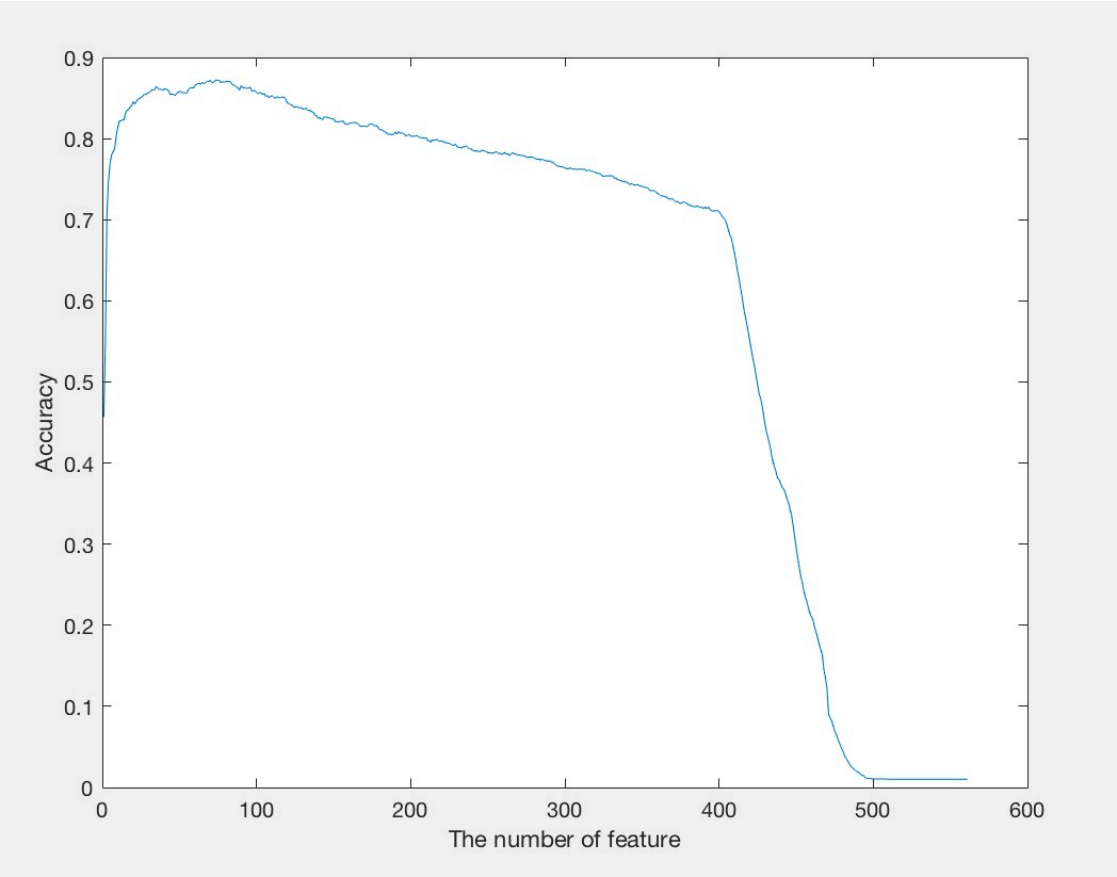
Data Set Characteristics:	Multivariate, Time-Series	Number of Instances:	7767
Attribute Characteristics:	Real	Number of Attributes:	561
Associated Tasks:	Classification	Missing Values?	No

III. Principle Component Analysis

Due to the large number of features, I tried to implement the Principle Component Analysis for the original dataset. Using the pca( ) function in Matlab, I can get a reconstructed training dataset, with a 7767*561 Matrix. Then I applied the same component model to the test data, and the reconstructed test dataset is also a 7767*561 matrix.

In my reconstructed matrix, one row stands for one observation and one column stands for a constructed feature. The importance of the feature decreases as the column increases, like the first column feature represents the most of all the data, the second column feature represents secondly most of all the data.

In order to know the best number of constructed features to fit the model, I implemented Bayes Classifier to see how accuracy changes as the number of features increases. And from the graph below, we can see that the accuracy at first increases and starts to decrease as the number of feature reaches 74. So I can know that when we chose the first 74 features in the reconstructed dataset, the accuracy of classifier is the highest.



IV. Classification Methods

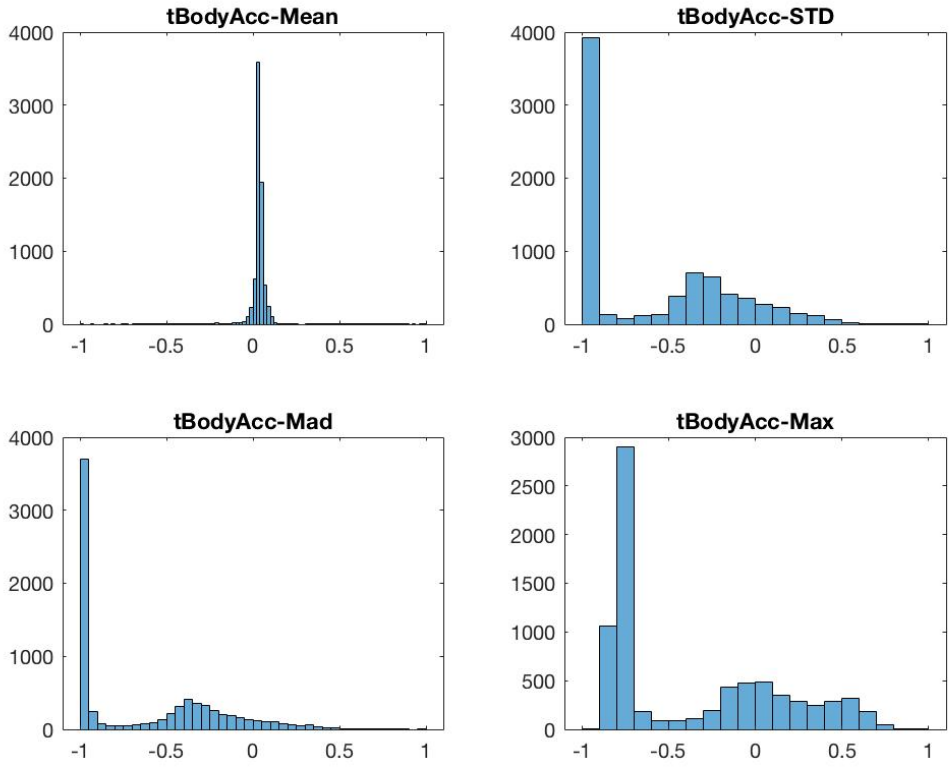
In this report, I use two classification methods to predict the labels. And for each classification, I use original dataset and reconstructed dataset separately and compare the accuracies between them. And I also adjust parameters for each of the classification and see the changes of accuracies.

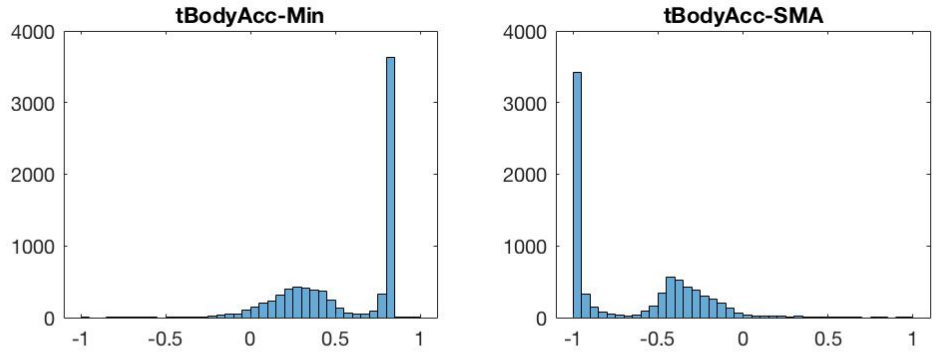
A.Bayes Classification

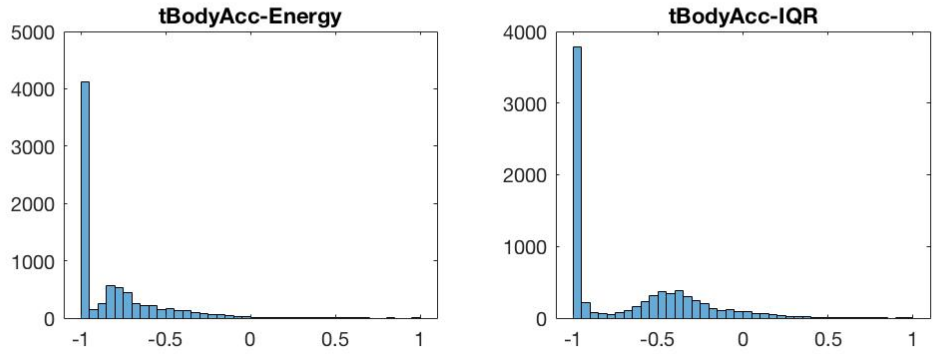
Bayes classifier is a probabilistic classifier based on applying Bayes' theorem with strong independence assumption between the features. In our dataset, I applied Bayes classification for original dataset and constructed dataset.

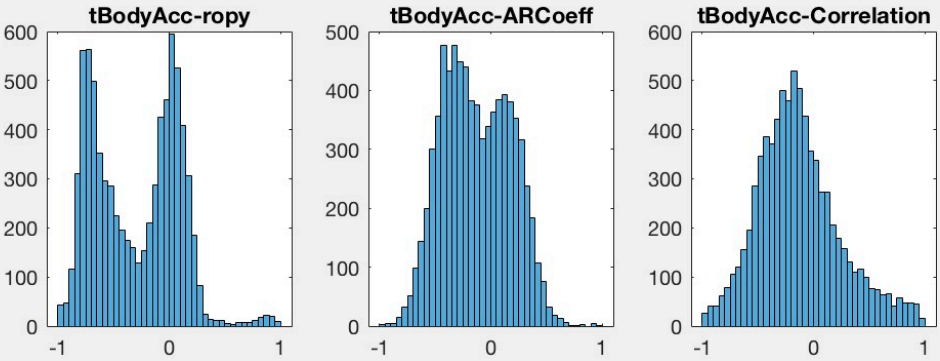
1.For original dataset

For the time saving purpose, I just choose tBodyAcc series to represent all the datasets, since it includes all of the 11 signal types (eg. Mean, std, max, min). This tBodyAcc is the first 40 features in our original dataset.









From the histogram graphs, we can see that there are 11 types of signals to estimate variables of the feature vector for each pattern. For tBodyAcc-Mean, tBodyAcc-ropy, tBodyAcc-ARCoeff, tBodyAcc-Correlation, they are nearly **Gaussian distribution**, for the other signals are nearly **exponential distribution.** In this case, I build a bayes classifier using different probability distribution based on the likelihood of each feature. Note that I used **Maximum A Posteriori** approach to determine the classes, and set the prior probability based on the class distribution in the training dataset.

Size of training data set	1153 Training data (20% of Total dataset)	3107 Training data (40% of Total dataset)	4660 Training data (60% of Total dataset)	4660 Training data (80% of Total dataset)	7764 Training data (100% of Total dataset)
Accuracy	0.5534	0.5882	**0.6199**	0.5718	0.5769

From the graph, we can see that when the size of training data increases, the accuracy of model at first increases and then decreases. When using 60 % training dataset, there is the highest accuracy 61.99%. The reason for this situation might be the high amount of training data at first trains the model accurately, but as the amount of training data increases, the model might be overfitting and lead to a lower test accuracy.

Prior Probability	Using the class distribution of training data	Using prior probability 25% for Class 1, the other is evenly distributed	Using prior probability 25% for Class 2, the other is evenly distributed	Using prior probability 25% for Class 3, the other is evenly distributed	Using prior probability 25% for Class 4, the other is evenly distributed
Accuracy	0.5769	0.5693	0.5648	0.5683	**0.5968**

Prior Probability	Using prior probability 25% for Class 5, the other is evenly distributed	Using prior probability 25% for Class 6, the other is evenly distributed	Using prior probability 25% for Class 7, the other is evenly distributed	Using prior probability 25% for Class 8, the other is evenly distributed	Not using Prior Probability
Accuracy	0.5440	0.5961	0.5699	0.5734	0.5743

The above graph shows how the accuracy changes as the prior probability changes. I choose 10 different values for the prior probability, including the probability distribution in training dataset, the prior probability which has more weight on a specific class, and not using prior probability. And before experimentation, I guess that the accuracy will be highest when using the probability distribution in training dataset.

But as we can see that the accuracy does not change too much whether using prior probability or not. The accuracy is not highest when choosing probability distribution in training dataset as prior probability, the highest accuracy is **59.68%** when I put more prior probability for class 4. This might prove that Class 4 are more likely to happen than other classes in testing our datasets.

2. For Dimension reduced data

In this step, I use the data after PCA to build Bayes Classifier. Note that after PCA, every feature in the reconstructed dataset is **Gaussian** distribution, so I just use Gaussian distribution to fit the likelihood of every feature. I will also try two hyper-parameters to build the model: The size of training data and prior probability.

1.The size of training data

Size of training data set	1153 Training data (20% of Total dataset)	3107 Training data (40% of Total dataset)	4660 Training data (60% of Total dataset)	4660 Training data (80% of Total dataset)	7764 Training data (100% of Total dataset)
Accuracy	0.8147	0.8450	0.8643	**0.8722**	**0.8722**

From the graph, we can see that the accuracy of the model greatly increases after we implement PCA. And as the size of training data increases, the accuracy continuously increases. The highest accuracy is **87.22%** when we use 80% or full dataset.

2. Prior probability

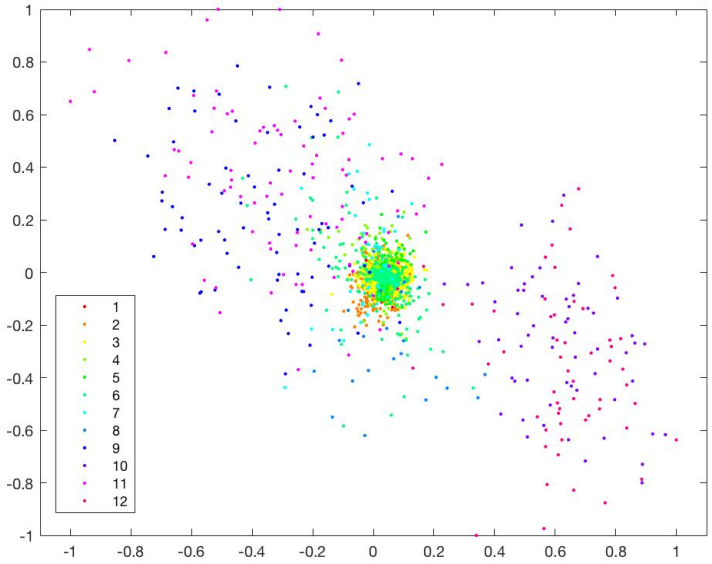
Prior Probability	Using the class distribution of training data	Using prior probability 25% for Class 1, the others are evenly distributed	Using prior probability 25% for Class 2, the others are evenly distributed	Using prior probability 25% for Class 3, the others are evenly distributed	Using prior probability 25% for Class 4, the others are evenly distributed
Accuracy	0.8722	0.8586	0.8707	0.8713	**0.8744**

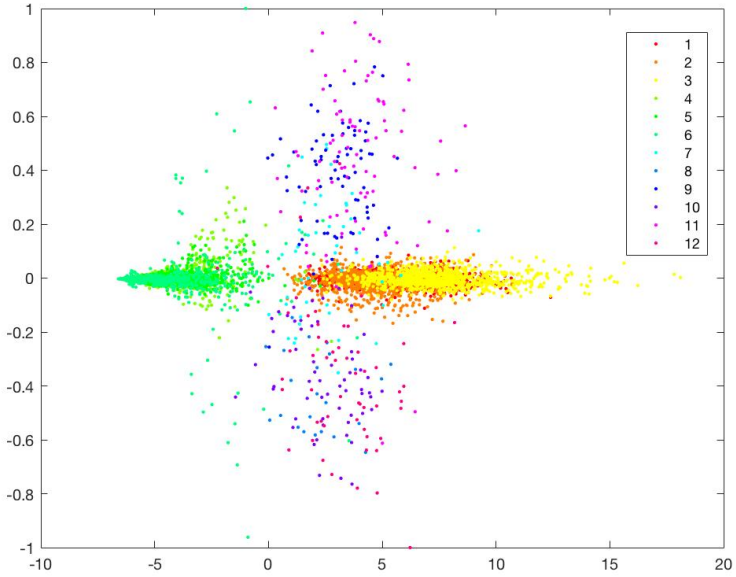
Prior Probability	Using prior probability 25% for Class 5, the others are evenly distributed	Using prior probability 25% for Class 6, the others are evenly distributed	Using prior probability 25% for Class 7, the others are evenly distributed	Using prior probability 25% for Class 8, the others are evenly distributed	Not using Prior Probability
Accuracy	0.8627	0.8722	0.8710	0.8722	0.8732

As we can see in the graph above, I still choose 10 different values for the prior probability. And the accuracy is the highest when I put more prior probability on the class 4. This is the same as it is for the original dataset. So we can conclude that the class 4 should have higher prior probability than other classes.

B.SVM

Support vector machines are supervised learning models that analyze data used for classification. I use SVM for original data and dimension-reduced dataset and analyze the accuracies using two different hyper-parameter values.





These two graphs are the classes distribution in 2D graph using the first two features as example. The first graph is the data distribution for the original dataset, and the second graph is the data distribution for the dimension-reduced dataset. As we can see that the data in dimension-reduced dataset is more disperse and distinguishable than the data in original dataset.

1.Kernel Type

Parameter	C=1.0 Kernel=RBF	C=1.0 Kernel=Linear	C=1.0 Kernel=poly	C=1.0 Kernel=sigmoid
Accuracy of the original dataset	0.9181	**0.9519**	0.8728	0.8823
Accuracy of the dimension reduced dataset	**0.9285**	0.9225	0.3893	0.9256

From the graph, we can see that when C value is given, as the Kernel type changes, the accuracy changes a lot. For the original dataset, the highest accuracy is **95.19%** when Kernel type is Linear. For the dimension-reduced dataset, the highest accuracy is **92.85%** when Kernel type is RBF. (I just choose four values since there are just four Kernel Type in SVM)

2.C value

Parameter	C=0.1 Kernel=RBF	C=0.5 Kernel=RBF	C=1.0 Kernel=RBF	C=2.0 Kernel=RBF	C=5.0 Kernel=RBF
Accuracy of the original dataset	0.8124	0.8921	0.9181	0.9294	**0.9433**
Accuracy of the dimension reduced dataset	0.8871	0.9247	0.9285	0.9320	**0.9345**

As we can see that the accuracy increases as C value increases. For original dataset, the highest accuracy is **94.33%,** for dimension-reduced dataset, the highest accuracy is **93.45%.** When C value is small, it has a large margin which might misclassify more data. And when C is big, it has a smaller margin and will fit better to the training model. So that might be the reason why a higher C value leads to a higher accuracy.

V. Conclusion

To conclude, this project includes Principle Component Analysis, Bayes Classifier regarding Gaussian and Exponential distribution and Support Vector Machines.

For Bayes Classifier, there are three conclusions from my experimentation:

1) The accuracy of dimension-reduced dataset is much higher than that of original dataset (Maybe because I just select 40 features for the original dataset).
2) The accuracy does not always increase as the training data size increases. Sometimes more training data will lead to overfitting.
3) It is not the best to choose the probability distribution of the training dataset as prior probability. In this dataset, when we add more probability weight to class 4, the accuracy will be higher.

For Support Vector Machines:

1) The accuracy of dimension-reduced dataset is slightly higher than that of original dataset.
2) As C value increases, the accuracy increases. (But large C value might cause a overfitting)
3) For original dataset, the accuracy is the highest when its kernel type is Linear. For dimension dataset, the accuracy does not change too much in different kernel types, while it has a bad accuracy in Poly kernel type.

In conclusion, the performance of SVM is better than Bayes Classifier. I get the highest accuracy in this project is 95.19% using SVM (Original dataset, C=1.0, Kernel=linear)