University of Virginia

Project for CPE 7993: Independent Study

Recognition of Human Activities

Zhidan Luo

November 2019

1 Introduction

1.1 Background

A current engineering challenge is to identify human activity (e.g., walking, in car, on bike, eating, smoking, falling) from smartphones and other wearable devices. More specifically, the embedded sensors (e.g., accelerometers and gyroscopes) produce a time series of position, velocity, and acceleration measurements. These time series are then processed to produce a set of features that can be used for activity recognition. The aim of this topic is to develop a supervised method to classify observations into human activities categories.

1.2 About data

The dataset is a public domain dataset for human activity recognition using smartphones. The details of the data collection process and features can be found in this paper. There are 6 classes in the dataset:

Label	Description
0	Walking
1	W. upstairs
2	W. downstairs
3	Sitting
4	Standing
5	Laying down

Table 1: About the Data

The first column are the labels and the remaining columns are the 561 predictor variables. The dataset has been also randomly partitioned into two independent sets, where 70% of the data were selected for training and the remaining 30% for testing.

2 Proposed Model

2.1 Support Vector Machine (SVM)

SVM is a type of generalized linear classifier of supervised learning that classifies data in a binary manner. The decision boundary is the maximum-margin hyperplane for solving learning samples. SVM uses hinge loss to calculate empirical risk and adds regularization terms to the solution system to optimize structural risk. It is a classifier with sparsity and robustness. SVM can perform non-linear classification by kernel method, which is one of the common kernel learning methods.
In this project, two important parameters of SVM have been considered.

- C: penalty parameter of the error term, choosing from $\{0.1, 1, 10\}$.
- kernel: specified kernel type to be used in the algorithm, choosing from {'linear', 'poly', 'rbf', 'sigmoid'}. Additionally, degree d of the polynomial kernel function were considered, choosing from $\{1, 3, 5, 7\}$.

2.2 Principal Component Analysis (PCA)

PCA is a common method of data analysis. The original data is linearly transformed into a set of representations of which each dimension is linear independent. This can be used to extract the main feature components of the data. PCA is often used to reduce the dimension of high dimensional data.
In this project, there are 561 features. 10 different number of principal components have chosen including the original non-dimension-reduced dataset to obtain the best test accuracy. That is,

$$\text{nComponentsList} = [56, 112, 168, 224, 280, 336, 392, 448, 504, 561]$$

2.3 K-fold Cross Validation

The dataset is divided into K sub-samples. A single sub-sample is retained as the data for the testing. The other $K-1$ samples are used for training. The cross-validation is repeated K times, and each sub-sample is tested once. The results of the average K times or other combination methods are used to finally obtain a single estimate.
The advantage of this method is that it repeatedly uses randomly generated sub-samples for training and testing, and the results are verified once each time.
In this project, $K = 4$ was utilized to obtain the accuracy for training the SVM model each time.

2.4 Grid Search

Grid search is one of the methods to optimize parameters. It is a kind of exhaustive search. In all candidate parameter selections, through looping through and trying every possibility, the best performing parameter is the final result. The principle is like finding the maximum value in an array.
Take a SVM model with two parameters as an example. As described in section 2.1, Parameter C has 3 possibilities and parameter 'kernal' has 4 possibilities. List all the possibilities and it can be expressed as a 3×4 table as shown in table 2, where each cell is a grid. The looping process is to search in each grid, so it is called grid search.

	$C = 0.1$	$C = 1$	$C = 10$
linear	SVC($C = 0.1$, kernel = 'linear')	SVC($C = 1$, kernel = 'linear')	SVC($C = 10$, kernel = 'linear')
poly	SVC($C = 0.1$, kernel = 'poly')	SVC($C = 1$, kernel = 'poly')	SVC($C = 10$, kernel = 'poly')
rbf	SVC($C = 0.1$, kernel = 'rbf')	SVC($C = 1$, kernel = 'rbf')	SVC($C = 10$, kernel = 'rbf')
sigmoid	SVC($C = 0.1$, kernel = 'sigmoid')	SVC($C = 1$, kernel = 'sigmoid')	SVC($C = 10$, kernel = 'sigmoid')

Table 2: Grid Search Example

3 Results

The results of loss and accuracy for different numbers of principal components are shown in Table 3.

N Components	Training Loss	Training Accuracy	Testing Loss	Testing Accuracy
56	0.009929	0.990071	0.0580	0.9420
112	0.007889	0.992111	0.0512	0.9488
168	0.007481	0.992519	0.0383	0.9617
224	0.006937	0.993063	0.0373	0.9627
280	0.006529	0.993471	0.0360	0.9640
336	0.006393	0.993607	**0.0353**	**0.9647**
392	0.006121	0.993879	0.0360	0.9640
448	0.006121	0.993879	0.0360	0.9640
501	0.006121	0.993879	0.0360	0.9640
561	0.006121	0.993879	0.0360	0.9640

Table 3: Results for Different Components

Besides, the training loss and accuracy as well as testing loss and accuracy vs different numbers of principal components are shown figure 1 and 2 respectively.

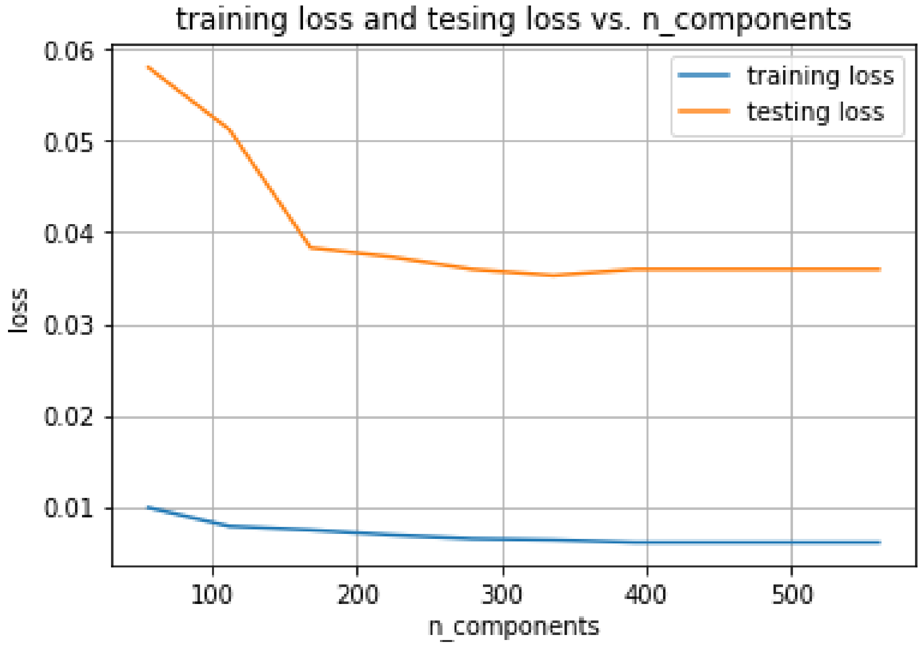

Figure 1: Training Loss and Accuracy vs N Components

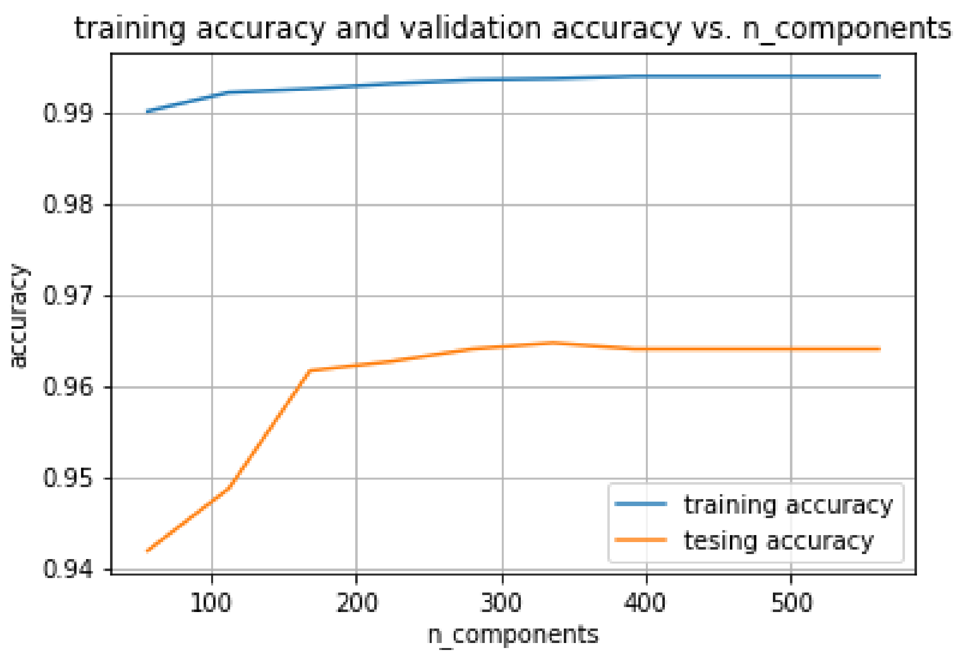

Figure 2: Testing Loss and Accuracy vs N Components