code book

CodeBook.md describes the variables, the data, and any transformations or work that I have performed to clean up the data

Source:

Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto.

Smartlab - Non Linear Complex Systems Laboratory

DITEN - UniversitÃ degli Studi di Genova, Genoa I-16145, Italy.

activityrecognition '@' smartlab.ws

www.smartlab.ws

Data Set Information:

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, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Check the README.md file for further details about this dataset.

Attribute Information:

For each record in the dataset it is provided:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.

- Triaxial Angular velocity from the gyroscope.

- A 561-feature vector with time and frequency domain variables.

- Its activity label.

- An identifier of the subject who carried out the

experiment.

Activity labels:

1 WALKING

2 WALKING_UPSTAIRS

3 WALKING_DOWNSTAIRS

4 SITTING

5 STANDING

6 LAYING

feature vector, variables:

#1 tBodyAcc-mean()-X #2 tBodyAcc-mean()-Y #3 tBodyAcc-mean()-Z #4 tBodyAcc-std()-X #5 tBodyAcc-std()-Y #6 tBodyAcc-std()-Z #7 tBodyAcc-mad()-X #8 tBodyAcc-mad()-Y #9 tBodyAcc-mad()-Z #10 tBodyAcc-max()-X #11 tBodyAcc-max()-Y #12 tBodyAcc-max()-Z #13 tBodyAcc-min()-X #14 tBodyAcc-min()-Y #15 tBodyAcc-min()-Z #16 tBodyAcc-sma() #17 tBodyAcc-energy()-X #18 tBodyAcc-energy()-Y

and so on …

#511 fBodyAccMag-entropy() #512 fBodyAccMag-maxInds #513 fBodyAccMag-meanFreq() #514 fBodyAccMag-skewness() #515 fBodyAccMag-kurtosis() #516 fBodyBodyAccJerkMag-mean() #517 fBodyBodyAccJerkMag-std() #518 fBodyBodyAccJerkMag-mad() #519 fBodyBodyAccJerkMag-max() #520 fBodyBodyAccJerkMag-min() #521 fBodyBodyAccJerkMag-sma() #522 fBodyBodyAccJerkMag-energy() #523 fBodyBodyAccJerkMag-iqr() #524 fBodyBodyAccJerkMag-entropy() #525 fBodyBodyAccJerkMag-maxInds

…

#560 angle(Y,gravityMean) #561 angle(Z,gravityMean)

complete list could be found in feature.txt

Data cleaning and transformation is decribed in README.md, I will only briefly repeat

some of the information:

download data from

http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+

unzip it at your working directory

it will create directoy called “UCI HAR Dataset”

in that directory there are two other directories: “test” and “train”,

as well as labels of variour columns of the data in txt files

create a data frame “subject” by reading subject_test.txt, which has rows of a subject's number,

and name the column “subject”

create a data frame “activity” by reading y_test.txt, which has rows of an activity's number,

and name the column “activity”

change the numbers in activity data frame into textual activities listed in “activity_labels.txt”

merge two data frames into one (subject and activity into dat1)

read X_test data with 561 columns

get the names of the features (which will be column names of X_test data)

rows should be columns, so we will transpose rows and columns, and get just the names of the features

rename the columns of X_test data with the column names of the features from features.txt

get the index of columns that have “mean()” in them, and also “meanFreq()

get the index of columns that have "std()” in them

combine those indices into one

sort the indices

these are the indices we need in our data

the data set with columns of needed data of mean, and std computations

merge subject + activity (dat1), and X_test_mean_std

just to check the final result: nrow(X_test_merged) gives me 2947,

ncol(X_test_merged) gives me 81, so the test merged data frame is 2947 x 81

Do the same for the train data

just to check the final result: nrow(X_train_merged) gives me 7352,

ncol(X_train_merged) gives me 81, so the train merged data frame is 7352 x 81

merging the test and the training sets should give me (2947 + 7352) x 81 data frame

i.e. 10299 x 81

nrow(X_final) gives me 10299, which is as expected, ncol(X_final) is 81 as well

empty data frame for tidy data for future storage there

the data is split by subject, activity, and other variables, so it computes

the mean of each variable for each subject and his/her activity,

e.g. subject 1, activity 1, all means of variables; subject 2, activity 1, means,

up to last subject, activity 1

then, it starts with subject 1, activity 2 …, last s, act 2 and so on

nrow(X_tidy) = 79, ncol(X_tidy) = 180

output with row.names=F so the data is properly aligned in MS Excel, but the row names

are, of course, missing

output with row.names=T so the data is NOT properly aligned in MS Excel