FFT Features Extraction {Code Documentation}

Rischan Mafrur

Monday, December 15, 2014

To extract the FFT coefficients from our dataset we use the similar code with the previouse one (time domain features extraction). The different is starting after we extract signals to the gait cycles. In the time domain features, after we extract the data to the gait cycles we define the function to extract time features but in this case we define function to extract FFT coefficients, the code looks like: (* do not forget to load `rwt` and `fftw` libraries).

```
library('rwt')
library('fftw')
  #This code was created by Alvin Prayuda, modified by Rischan Mafrur
  #40 FFT Coefficient
  fft_coef_all <- data.frame()
  for  (counter in 1:total_gait_cycle)
  {
    temp1 <- paste("gait_cycles_Y",counter,sep = "_")
    temp2 <- (get(temp1)$Index[length(get(temp1)$Index)]-get(temp1)$Index[1])*time_sca
le
    time_gap <- temp2

    temp1 <- paste("gait_cycles_X",counter,sep = "_")
    temp2 <- get(temp1)$Value
    temp_func <- approxfun(1:length(temp2),temp2,method='linear')
    temp3 <- matrix(temp_func(seq(1,length(temp2),length.out=256)),1,length(temp_func
(seq(1,length(temp2),length.out=256))))
    fft_plan <- planFFT(length(temp3))
    fft_x <- FFT(temp3,plan=fft_plan)
    re_coef_fft_x <- matrix(Re(fft_x[1:40]),1,length(Re(fft_x[1:40])))

    temp1 <- paste("gait_cycles_Y",counter,sep = "_")
    temp2 <- get(temp1)$Value
    temp_func <- approxfun(1:length(temp2),temp2,method='linear')
    temp3 <- matrix(temp_func(seq(1,length(temp2),length.out=256)),1,length(temp_func
(seq(1,length(temp2),length.out=256))))
    fft_plan <- planFFT(length(temp3))
    fft_y <- FFT(temp3,plan=fft_plan)
    re_coef_fft_y <- matrix(Re(fft_y[1:40]),1,length(Re(fft_y[1:40])))


    temp1 <- paste("gait_cycles_Z",counter,sep = "_")
    temp2 <- get(temp1)$Value
    temp_func <- approxfun(1:length(temp2),temp2,method='linear')
    temp3 <- matrix(temp_func(seq(1,length(temp2),length.out=256)),1,length(temp_func
(seq(1,length(temp2),length.out=256))))
    fft_plan <- planFFT(length(temp3))
    fft_z <- FFT(temp3,plan=fft_plan)
    re_coef_fft_z <- matrix(Re(fft_z[1:40]),1,length(Re(fft_z[1:40])))

    temp1 <- paste("gait_cycles_M",counter,sep = "_")
    temp2 <- get(temp1)$Value
    temp_func <- approxfun(1:length(temp2),temp2,method='linear')
    temp3 <- matrix(temp_func(seq(1,length(temp2),length.out=256)),1,length(temp_func
```

```
(seq(1,length(temp2),length.out=256))))
    fft_plan <- planFFT(length(temp3))
    fft_m <- FFT(temp3,plan=fft_plan)
    re_coef_fft_m <- matrix(Re(fft_m[1:40]),1,length(Re(fft_m[1:40])))

    temp_coef <- cbind(time_gap,re_coef_fft_x,re_coef_fft_y,re_coef_fft_z,re_coef_fft_
m)
    temp_coef <- as.data.frame(temp_coef)
    fft_coef_all <- rbind(fft_coef_all,temp_coef)
  }

  extracted_feature <- rbind(extracted_feature,fft_coef_all)
}
```

To see how many rows and how many columns in our result, we can use nrow and ncol. The important thing is ncol.

```
nrow(extracted_feature) #number of rows of our data

## [1] 204

ncol(extracted_feature) #number of columns of our data

## [1] 161
```

Based on ncol, we have 161 columns. It means 160 FFT coefficents + 1 label. 160 FFT coefficients came from 40 FFT coefficients from each signals. In this implementation we use all of signals, the total is 4 signals (X,Y,Z, and M).

Try to see the top from FFT data.

```
extracted_feature[1:5,1:5] # only show first 5 rows and columns

##   time_gap        V2         V3          V4          V5
## 1 243.0234  12.92793 -10.104905  -3.0097463 -0.07133957
## 2 405.0391  26.80716   1.737334  -8.6580186  1.26671131
## 3 182.2676  96.07463  89.670503 -49.2316767  1.64050747
## 4 121.5117 154.20093  -4.587178   0.9776246 -0.45388777
## 5 101.2598 106.16043  -3.204524   2.1830050  1.14983574
```

We store all of variables in CSV files. If we need to process these data just load the CSV files.