



UNIVERSIDADE D
COIMBRA

Relatório do Projeto de ATD

Trabalho realizado por:

Afonso Gouveia e Melo 2020217360

Eduardo José Nunes 202017675

Tomás Logrado Tavares 2020225823

Introdução

Este projeto tem como objetivo a análise de dados, nos domínios do tempo e da frequência, usando sinais recolhidos de acelerómetros de smartphones e utilizando os conhecimentos aprendidos na disciplina de Análise e Transformação de Dados.

Resumo do projeto:

Começámos o nosso trabalho por fazer download dos sinais, e importámos o seu conteúdo através da função “importdata” do MatLab.

Foram criadas 4 arrays, com as experiências, utilizadores, sensores e atividades, respetivamente, para criar todas as figuras de uma vez.

Para a criação das labels utilizámos, também, a função “importdata” no ficheiro “labels.txt”.

```
%% Exercicio 1
% Inicializar variaveis

Fs = 50; % Frequencia de amostragem dada no enunciado - passos por minuto
exp= {"34","35","36","37","38","39","40","41"}; % Array para os exp para os ficheiros
User = {"17","18","19","20"}; % Array para os users para os ficheiros

% Array com todas as atividades para conseguirmos dar label
atividades = {"W","WU","WD","S","ST","L","STSit","SitTS","SitTL","LTSit","STL","LTS"};
% Array para as legendas dos graficos
sensores = {"ACC_X","ACC_Y","ACC_Z"};

% imports:
labels_info = importdata("HAPT Data Set/RawData/labels.txt");
```

Ao carregar os ficheiros, foi criado um loop de maneira a poder associar cada utilizador, a duas experiências. Especificamente associámos o utilizador 17 à experiência 34, e o utilizador 20, à experiência 41. Com isto obtemos os dados dos acelerómetros das respetivas experiências e, depois disso, o gráfico das diferentes atividades nas 3 dimensões.

```

for k=1:numel(exp)
    if (par==1) % cada user tem duas exp
        users = users + 1; % se ja tiver passado uma vez, aumenta o user.
        par=0;
    else
        % se nao continua o mesmo user
        par = 1;
    end

    % carregar ficheiro correspondente aos valores obtidos
    ficheiro = sprintf("HAPT Data Set/RawData/acc_exp%s_user%s.txt",exp{k},User{users})
    data_read = importdata(ficheiro);
    data = data_read;
    [n_pontos,n_plots] = size(data); % tamanho da data

    % labels para o ficheiro correspondente
    info_labels = intersect(find(labels_info(:,1)== str2num(exp{k})),find(labels_info(:,2)== str2num(User{users})));

    % Criar vetor da data de acordo com a frequencia de amostragem obtida
    t=[0:size(data,1)-1]./Fs;

    % Desenha 8 figuras independentes
    figure;

    for i=1:n_plots % Desenha o grafico em si, 3 por figura
        subplot(n_plots,1,i);plot(t./60,data(:,i),"k--")
        axis([0 t(end)./60 min(data(:,i)) max(data(:,i))])
        xlabel("Time(mins)","fontsize",16,"fontweight","bold")
        ylabel(sensores{i},"fontsize",16,"fontweight","bold");
        hold on % salva o plot atual
        for j=1:numel(info_labels)
            plot(t(labels_info(info_labels(j),4):labels_info(info_labels(j),5))./60,
                data(labels_info(info_labels(j),4):labels_info(info_labels(j),5),i))

            if (mod(j,2)) == 1 % vai ver se deve ser em cima ou em baixo
                ypos = min(data(:,i))-(0.2*min(data(:,i)));
            else
                ypos = max(data(:,i))-(0.2*max(data(:,i)));
            end
            % aqui escreve efetivamente as labels
            text(t(labels_info(info_labels(j),4))/60,
                ypos,
                atividades{labels_info(info_labels(j),
                3)},"VerticalAlignment","top","HorizontalAlignment","left");
        end
    end
end
end

```

Com isto, avançamos para o exercício 3.

Neste exercício começamos por criar os 12 casos diferentes para cada atividade e a data que cada um destes casos vai ter. Depois disso, calculamos o DFT de cada uma das atividades dinâmicas juntamente com a sua amplitude, especificidade e sensibilidade.

Durante as atividades dinâmicas o número de passos por minuto de cada utilizador ia sendo guardando para posteriormente criar uma tabela com o valor médio e o desvio padrão por atividade.

Finalmente, para terminar o exercício 3, calculamos as diferentes médias e os desvios padrões.

```

%% Exercício 3
% Arrays com os passos por segundo
steps_w = [];
steps_wup = [];
steps_wdown = [];

%Passos Totais (Uma parcela é um ficheiro)
total_steps_w = zeros(numel(exp),1);
total_steps_wup = zeros(numel(exp),1);
total_steps_wdown = zeros(numel(exp),1);

% foi usada a janela "blackman"
for n=1:numel(exp) % Para as atividades dinamicas:
    %WALKING
    for i=1:size(w_z{n},2)
        black_win = blackman(numel(w_z{n}{i}));
        dft = fftshift(fft(detrend(w_z{n}{i}).*black_win));
        m = abs(dft);

        f = (0:length(dft)-1)*100/length(dft);

        [pks_m, pks_f] = findpeaks(m);
        Ts=1/pks_f(1);

        steps_w = [steps_w; 60/Ts];
        total_steps_w(n) = total_steps_w(n) + (size(w_z{n}{i},1)/Fs)/Ts;
    end

    %WALKING UPSTAIRS
    for i=1:size(w_up_z{n},2)
        black_win = blackman(numel(w_up_z{n}{i}));
        dft = fftshift(fft(detrend(w_up_z{n}{i}).*black_win));
        m = abs(dft);
        f = (0:length(dft)-1)*100/length(dft);

        [pks_m, pks_f] = findpeaks(m);

        Ts=1/pks_f(1);

        steps_wup = [steps_wup; 60/Ts];
        total_steps_wup(n) = total_steps_wup(n) + (size(w_up_z{n}{i},1)/Fs)/Ts;
    end

    %WALKING DOWNSTAIRS
    for i=1:size(w_down_z{n},2)
        black_win = blackman(numel(w_down_z{n}{i}));
        dft = fftshift(fft(detrend(w_down_z{n}{i}).*black_win));
        m = abs(dft);
        f = (0:length(dft)-1)*100/length(dft);

        [pks_m, pks_f] = findpeaks(m);

        Ts=1/pks_f(1);

        steps_wdown = [steps_wdown; 60/Ts];
        total_steps_wdown(n) = total_steps_wdown(n) + (size(w_down_z{n}{i},1)/Fs)/Ts;
    end
end

% Media steps por minuto
media_steps_w_sec = mean(steps_w)
media_steps_wup_sec = mean(steps_wup)
media_steps_wdown_sec = mean(steps_wdown)

% Desvio
dp_steps_w_sec = std(steps_w)
dp_steps_wup_sec = std(steps_wup)
dp_steps_wdown_sec = std(steps_wdown)

```

O exercício 4 foi num ficheiro à parte, visto que este servia apenas de exemplo para um data-set único. Aqui foi desenvolvida a nossa função de STFT, sendo esta aplicada a uma janela deslizante selecionada, hamming.



```
function [STFT, f, t] = stft(x, win, hop, nfft, fs)
% parse signal as a column vector
x = x(:);

% determination of the window length
wlen = length(win);

% stft matrix size estimation and preallocation
NUP = ceil((1+nfft)/2); % calculate the number of unique fft points
L = 1+fix((length(x)-wlen)/hop); % calculate the number of signal frames
STFT = zeros(NUP, L); % initialize the STFT matrix

% STFT (via time-localized FFT)
for l = 0:L-1
    % windowing
    xw = x(1+l*hop : wlen+l*hop).*win;
    % FFT
    X = fft(xw, nfft);
    % update of the stft matrix
    STFT(:, 1+l) = X(1:NUP);
end

% calculation of the time and frequency vectors
t = (wlen/2:hop:wlen/2+(L-1)*hop)/fs;
f = (0:NUP-1)*fs/nfft;

end
```