EEG Lab Assignment Project Report

1. Task 1.1

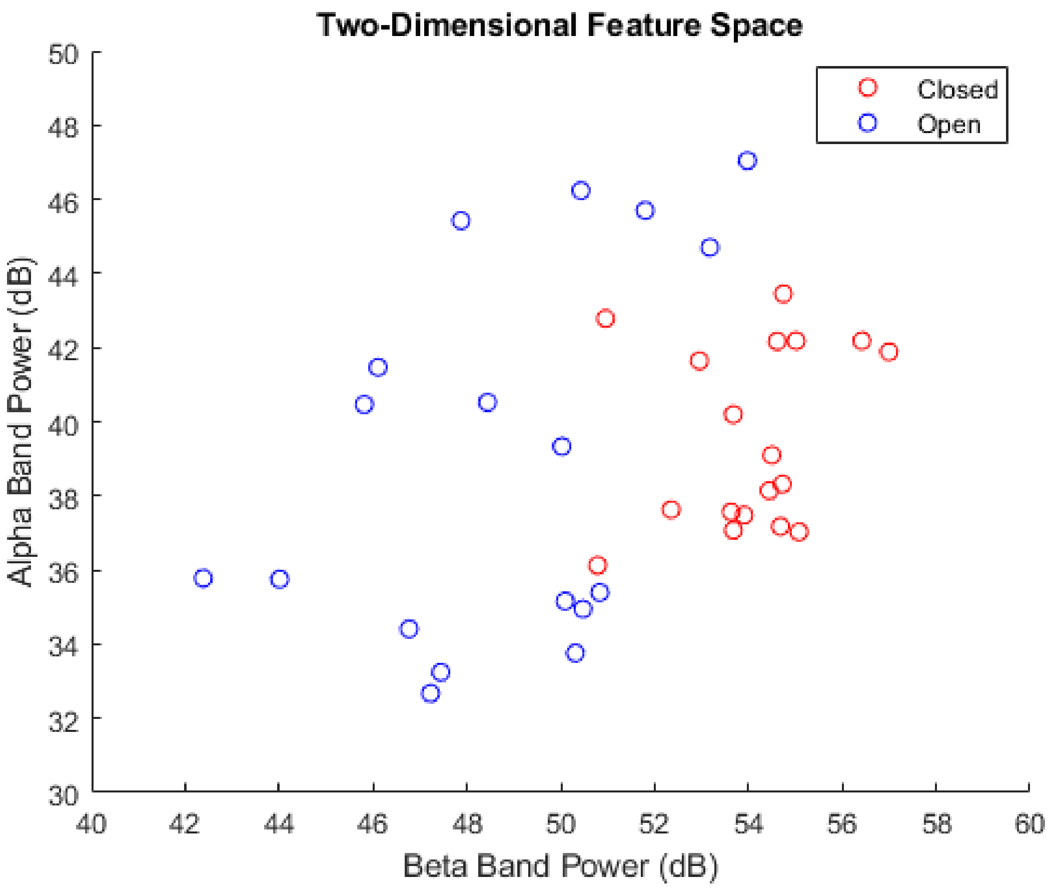

Figure 1 Indicates the mean alpha (8-12Hz) and beta (16-30Hz) band powers (in dB) relative to the DC gain of the captured signal (1.09×10^{-6}). The two-dimensional feature space above shows the relation between the features used to classify the data gathered at the beginning of the experiment. This plot enables us to visualise the relation between our two selected features, and apply a classification algorithm to classify the data points into their predicted categories (i.e. eyes open and eyes closed). To produce the data above, it was captured over three minutes in five-second intervals. The data was then run through a Fourier transform from which only the amplitude was extracted. This was then converted to the power density within both frequency bands described above.

2. Tasks 1.2, 1.3 & 1.4

Given that the categorisation of the data is essentially a two-choice forced alternative (2CFA), we can surmise that the minimum success rate would be 50% (guess rate). Provided the model is working at far above both the guess rate and critical rate (75% halfway between guess rate and 100%), we can conclude that it works fairly well. The separation of the data - visualised by the decision boundary plotted, categorises a majority of points to their respective groupings, and though the data is not completely linearly separable, the model produces only 6 errors. To improve upon this classification error, the model could have an increased polynomial order. Moreover, we could gather more data to further refine the data set. We could trial the same subject for a longer period of time, invite more subjects to trial, or both.

The production of these errors is an expected outcome of the testing environment and procedure. Electromagnetic interference from surrounding circuitry and other electrical

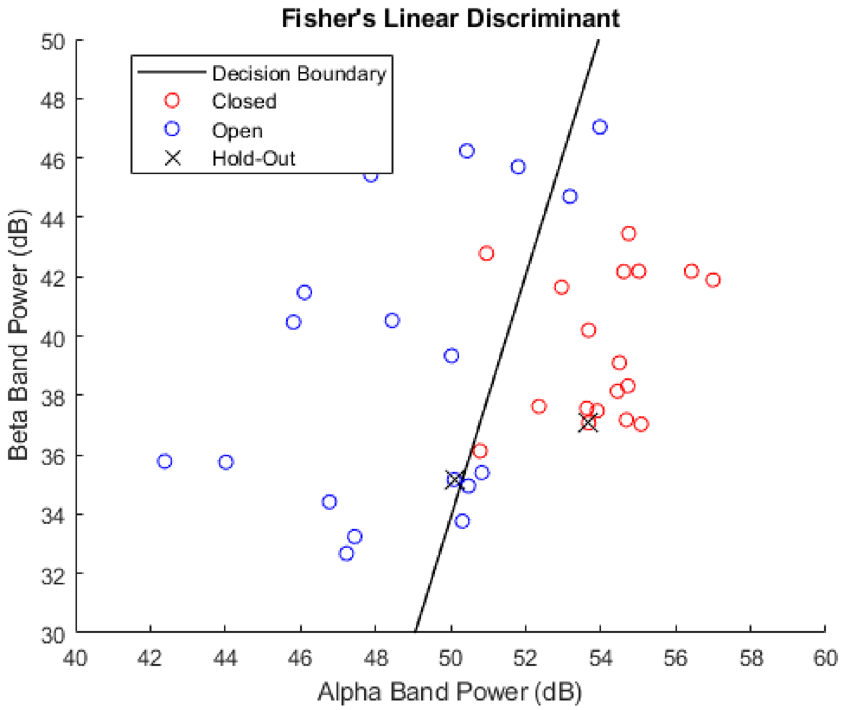

Figure 2 This is a plot of the aforementioned data. Superimposed are the results of one of an 18-fold cross-validation procedure using Fisher's Linear Discriminant classification method. Through cross-validation, I was able to find that the fitted model has an overall accuracy of approximately **80%**, miscategorising 6 of the 36 samples. It is also worth noting that the data without crosses overlayed is the hold-in data.

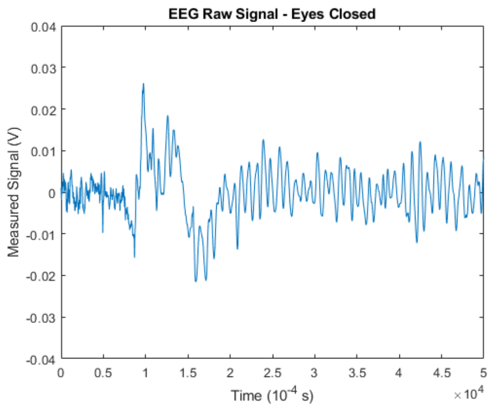

Figure 3 Data capture from T13, eyes closed

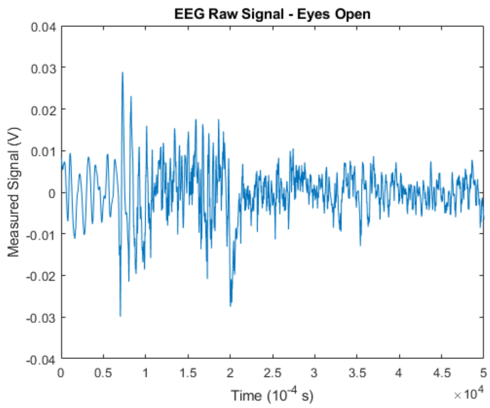

Figure 4 Data capture from T14, eyes open

instruments may have decreased the signal-to-noise ratio (SNR) of our data. Furthermore, alternating the tests within the trial of our subject may have produced a bleed of readings between tests. This is observed via a quick comparison of Figure 4 and Figure 3. The difference is apparent, with lower-frequency, higher-amplitude waves (alpha) being observed when the subject's eyes were closed - the inverse (beta) being true for when the subject's eyes were open. One may note the high-amplitude transitions between tests, as well as alpha waves parsing through into the "Eyes Open" test (Figure 4), and vice versa. To avoid the production of these outliers, we could have instead had the subject remain with their eyes closed for one half of the trial and eyes open for another. This would reduce the amount of potential bleed to between only two trials - the transition from eyes closed to eyes open. The results of this classifier are nevertheless positive and indicate fairly good categorisation throughout.

3. Task 1.5

A support vector machine was produced using a linear kernel, hard margin (no data fell within the margin region) and 18-fold cross-validation. Figure 5 indicates 11 support vectors were used and one can observe a steeper gradient (Figure 7 & Figure 6) in the decision boundary for the SVM as opposed to that of FLD's. This appears to create a better separation and performs better empirically. The success rate of this SVM was 86%. As compared to FLD, SVM performs better in this case. That said, a simple *ranksum* test (given the samples are not normally distributed with similar variance) proves that the difference between the two models is not statistically significant - producing a p-value of approximately 0.3. Performance could foreseeably be improved using a greater order polynomial kernel, and potentially softening the margin.

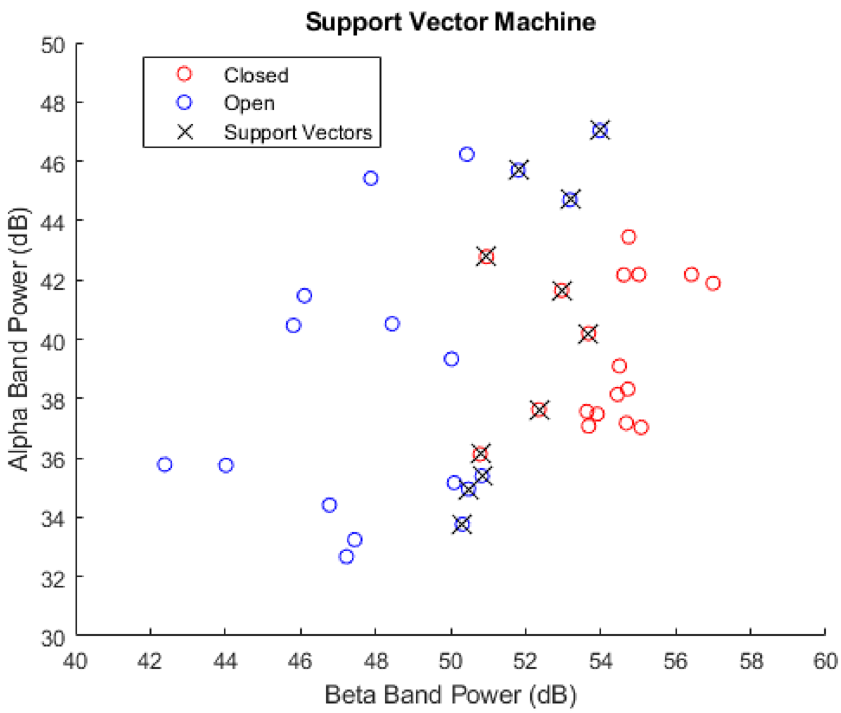

Figure 5 Support vector machine - resulting support vectors overlay on data seen before

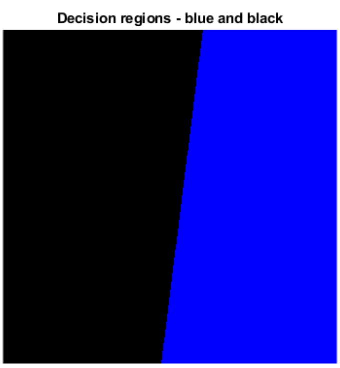

Figure 6 Support vector machine regions

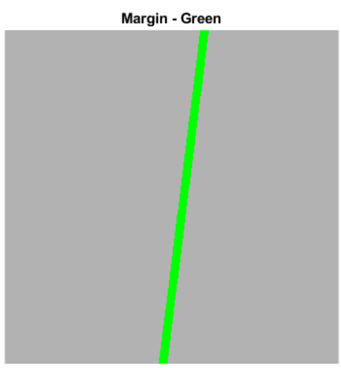

Figure 7 Support vector machine margin (hard)

4. Appendix

4.1 Tasks 1 and 2

```
% Clear workspace
clear
clc

%% Wrangle Data
% Number of trials in data
trials = 36;

% Load and shorten data
load("waveform.mat");
data = data(1:1800000); % clip data to correct length

% FFT signal
Fsig_ = fft(data) / length(data);
aFsig_ = abs(Fsig_);

% Split signal into trials
trial_data = cell(1,trials);
for i = 0:trials-1
    trial_data{i+1} = data(i*50000 + 1:(i+1)*50000)';
end

% Store power spectrum results in cell array
alpha = zeros(1,36);
beta = zeros(1,36);
for i = 1:length(trial_data)
    [pow_a, pow_b] = calculatePower(trial_data{i}, aFsig_(1));
    alpha(i) = pow_a;
    beta(i) = pow_b;

    % Create group vector
    if mod(i,2) == 1
        status(i) = "CLOSE";
        yn(i) = -1;
    else
        status(i) = "OPEN";
        yn(i) = 1;
    end

end

% Plot low-dim feature space
figure; hold on

plot(alpha(1:2:end), beta(1:2:end), "or");
plot(alpha(2:2:end), beta(2:2:end), "ob");

axis([40 60 30 50])

title("Two-Dimensional Feature Space")
xlabel('Beta Band Power (dB)')
ylabel('Alpha Band Power (dB)')

labels = ["Closed" , "Open"];
legend(labels, "Location","best");

%% Fisher's linear discriminant
```

```matlab
% Sort classifier hold-in and hold-out trials
power_band = [alpha', beta'];

% Result array
res = [];

% Classify data and test algo
for i = -1:2:trials-3
    train_data = power_band;
    trial_status = status;

    sample_inds = [(trials - i - 2) (trials - i - 1)];
    sample_data = power_band(sample_inds, :);

    train_data(sample_inds, :) = [];
    trial_status(sample_inds) = [];

    % Get decision border
    [C,err,posterior,logp,coeff] = classify(sample_data, train_data,
       trial_status);
    res = [res; C];

    K = coeff(1,2).const;
    L = coeff(1,2).linear;

    % Set up function
    f = @(alpha, beta) K + L(1)*alpha + L(2)*beta;

end

%% Plot results
% Plot decision boundary
figure; hold on
h3 = fimplicit(f);
h3.Color = 'k';
h3.LineWidth = 1;

plot(alpha(1:2:end), beta(1:2:end), "or");
plot(alpha(2:2:end), beta(2:2:end), "ob");
plot(sample_data(:, 1), sample_data(:,2), "xk", 'MarkerSize', 12);

axis([40 60 30 50])

title("Fisher's Linear Discriminant")
xlabel("Alpha Band Power (dB)")
ylabel("Beta Band Power (dB)")

labels = ["Decision Boundary", "Closed" , "Open", "Hold-Out"];

legend(labels, "Location","best");

hold off

res = [res, status'];

% Set up success rate
for i = 1:trials
    if (res(i,1) == res(i,2))
        res(i,3) = 1;
    else
```

```
            res(i,3) = 0;
        end
    end

%% Plot trial data for feature analysis
close all
figure;
t_sec = 1:50000;
plot(t_sec, trial_data{13})

title("EEG Raw Signal - Eyes Closed")
xlabel("Time (10^{-4} s)")
ylabel("Measured Signal (V)")

ylim([-0.04, 0.04])

figure;
t_sec = 1:50000;
plot(t_sec, trial_data{14})

title("EEG Raw Signal - Eyes Open")
xlabel("Time (10^{-4} s)")
ylabel("Measured Signal (V)")

ylim([-0.04, 0.04])

%% Post proc items for subsequent tasks
% Calc success rate
pct_succ = (1-length(find(res(:,3) == "0"))/trials) * 100;

power_band = [power_band, yn'];
```

4.2 Task 3

```
    % Get data from previous tasks
splitCalc;
DAT = power_band;

%% Set up SVM and classify
svm = fitcsvm(DAT(:,1:2), DAT(:,3), 'KernelFunction', 'linear');
crossval_svm = crossval(svm, 'kfold', 18);

% Calculate success rate
f = kfoldLoss(crossval_svm);
pct_succ(2) = (1 - f) * 100;

%% Plot SVM results
figure; hold on

plot(alpha(1:2:end), beta(1:2:end), "or");
plot(alpha(2:2:end), beta(2:2:end), "ob");

plot(svm.SupportVectors(:,1), svm.SupportVectors(:,2), 'xk', 'MarkerSize
    ', 12)

axis([40 60 30 50])

title("Support Vector Machine")
xlabel('Beta Band Power (dB)')
ylabel('Alpha Band Power (dB)')

labels = ["Closed" , "Open", "Support Vectors"];
legend(labels, "Location","best");

%% Plot SVM margin visualised
% Decision regions
[X1,X2] = meshgrid(linspace(0,100,1000));
X1 = X1(:);
X2 = X2(:);
[label, score] = predict(svm, [X1 X2]);
R = reshape(label, [1000 1000]);
figure; hold on
imagesc(R); axis image off; colormap([0 0 1; 0 0 0])
title('Decision regions - blue and black')

% Margin
R = reshape(score(:,2), [1000 1000]);
figure; hold on
imagesc(double((R >= -1) & (R <= 1)), [0 1]); axis image off; colormap
    ([0.7 0.7 0.7; 0 1 0])
title('Margin - Green')
```

```
% Convert predictions into success
FLD = cellfun(@str2num, res_fld(:,3));
SVM = cellfun(@str2num, res_svm(:,3));

% Two-sample ranksum test
p = ranksum(FLD,SVM);
```