

ECE 313 Final Project Report

Jacob Hutter, Junhao Pan

April 25, 2017

Task 0:

```
1  clear all;
   clc;
3
   load 1_a41178.mat;
5  patient1_data = floor(all_data);
   patient1_labels = all_labels;
7  training1 = patient1_data(:,1:(2*length(patient1_data)/3));
   testing1 = patient1_data(:,(2*length(patient1_data)/3):length(
   patient1_data));
9  label_training1 = patient1_labels(1:(2*length(patient1_labels)
   /3));
   label_testing1 = patient1_labels((2*length(patient1_labels)/3):
   length(patient1_labels));
11
   load 2_a42126.mat;
13 patient2_data = floor(all_data);
   patient2_labels = all_labels;
15 training2 = patient2_data(:,1:floor(2*length(patient2_data)/3))
   ;
   testing2 = patient2_data(:,floor(2*length(patient2_data)/3):
   length(patient2_data));
17 label_training2 = patient2_labels(1:floor(2*length(
   patient2_labels)/3));
   label_testing2 = patient2_labels(floor(2*length(patient2_labels)
   )/3):length(patient2_labels));
19
   load 3_a40076.mat;
21 patient3_data = floor(all_data);
   patient3_labels = all_labels;
23 training3 = patient3_data(:,1:(2*length(patient3_data)/3));
   testing3 = patient3_data(:,(2*length(patient3_data)/3):length(
   patient3_data));
25 label_training3 = patient3_labels(1:(2*length(patient3_labels)
   /3));
   label_testing3 = patient3_labels((2*length(patient3_labels)/3):
   length(patient3_labels));
27
   load 4_a40050.mat;
29 patient4_data = floor(all_data);
   patient4_labels = all_labels;
31 training4 = patient4_data(:,1:floor(2*length(patient4_data)/3))
   ;
```

```

testing4 = patient4_data(:, floor(2*length(patient4_data)/3):
length(patient4_data));
33 label_training4 = patient4_labels(1: floor(2*length(
patient4_labels)/3));
label_testing4 = patient4_labels( floor(2*length(patient4_labels
)/3): length(patient4_labels));
35
load 5_a41287.mat;
37 patient5_data = floor(all_data);
patient5_labels = all_labels;
39 training5 = patient5_data(:, 1:(2*length(patient5_data)/3));
testing5 = patient5_data(:, (2*length(patient5_data)/3): length(
patient5_data));
41 label_training5 = patient5_labels(1:(2*length(patient5_labels)
/3));
label_testing5 = patient5_labels((2*length(patient5_labels)/3):
length(patient5_labels));
43
load 6_a41846.mat;
45 patient6_data = floor(all_data);
patient6_labels = all_labels;
47 training6 = patient6_data(:, 1: floor(2*length(patient6_data)/3))
;
testing6 = patient6_data(:, floor(2*length(patient6_data)/3):
length(patient6_data));
49 label_training6 = patient6_labels(1: floor(2*length(
patient6_labels)/3));
label_testing6 = patient6_labels( floor(2*length(patient6_labels
)/3): length(patient6_labels));
51
load 7_a41846.mat;
53 patient7_data = floor(all_data);
patient7_labels = all_labels;
55 training7 = patient7_data(:, 1: floor(2*length(patient7_data)/3))
;
testing7 = patient7_data(:, floor(2*length(patient7_data)/3):
length(patient7_data));
57 label_training7 = patient7_labels(1: floor(2*length(
patient7_labels)/3));
label_testing7 = patient7_labels( floor(2*length(patient7_labels
)/3): length(patient7_labels));
59
load 8_a42008.mat;
61 patient8_data = floor(all_data);
patient8_labels = all_labels;
63 training8 = patient8_data(:, 1:(2*length(patient8_data)/3));
testing8 = patient8_data(:, (2*length(patient8_data)/3): length(
patient8_data));
65 label_training8 = patient8_labels(1:(2*length(patient8_labels)
/3));
label_testing8 = patient8_labels((2*length(patient8_labels)/3):
length(patient8_labels));
67
load 9_a41846.mat;
69 patient9_data = floor(all_data);
patient9_labels = all_labels;
71 training9 = patient9_data(:, 1: floor(2*length(patient9_data)/3))

```

```

;
testing9 = patient9_data(:, floor(2*length(patient9_data)/3):
length(patient9_data));
73 label_training9 = patient9_labels(1: floor(2*length(
patient9_labels)/3));
label_testing9 = patient9_labels( floor(2*length(patient9_labels
)/3): length(patient9_labels));

```

For task 0 we took the raw patient and gold data and partitioned it into testing and training segments.

Task 1.1:

```

HT_table_array_pat_1 = Get_HT_table(1, training1, label_training1);
2 HT_table_array_pat_2 = Get_HT_table(2, training2, label_training2);
HT_table_array_pat_3 = Get_HT_table(3, training3, label_training3);
4 HT_table_array_pat_4 = Get_HT_table(4, training4, label_training4);
HT_table_array_pat_5 = Get_HT_table(5, training5, label_training5);
6 HT_table_array_pat_6 = Get_HT_table(6, training6, label_training6);
HT_table_array_pat_7 = Get_HT_table(7, training7, label_training7);
8 HT_table_array_pat_8 = Get_HT_table(8, training8, label_training8);
HT_table_array_pat_9 = Get_HT_table(9, training9, label_training9);
10
HT_table_array = cat(1, HT_table_array_pat_1, HT_table_array_pat_2,
HT_table_array_pat_3, HT_table_array_pat_4,
HT_table_array_pat_5, HT_table_array_pat_6,
HT_table_array_pat_7, HT_table_array_pat_8,
HT_table_array_pat_9);

```

```

1 function HT_table_array_pat = Get_HT_table(patient_index,
patient_data, patient_labels)
3 % Prior probabilities, alarms/total
P_H1 = sum(patient_labels)/length(patient_labels);
5 P_H0 = 1 - P_H1;
7 HT_table_array_pat = cell(1, 7);
9 % name = strcat('Patient_', int2str(patient_index), '_Features');
% figure('name', name, 'unit', 'normalized', 'outerposition', [.1
.1 .8 .8]);
11
for i = 1:7
13 [feature_mat, x_mat] = Get_Feat_Mat(patient_data(i:i, :),
patient_labels);
15 ML_vector = zeros(1, length(feature_mat(1:1, :)));
MAP_vector = zeros(1, length(feature_mat(1:1, :)));
17
for k = 1:length(feature_mat(1:1, :))
19 P_H1_i = feature_mat(1, k);
P_H0_i = feature_mat(2, k);
21 if (P_H1_i >= P_H0_i)
% if H1_pmf >= H0_pmf

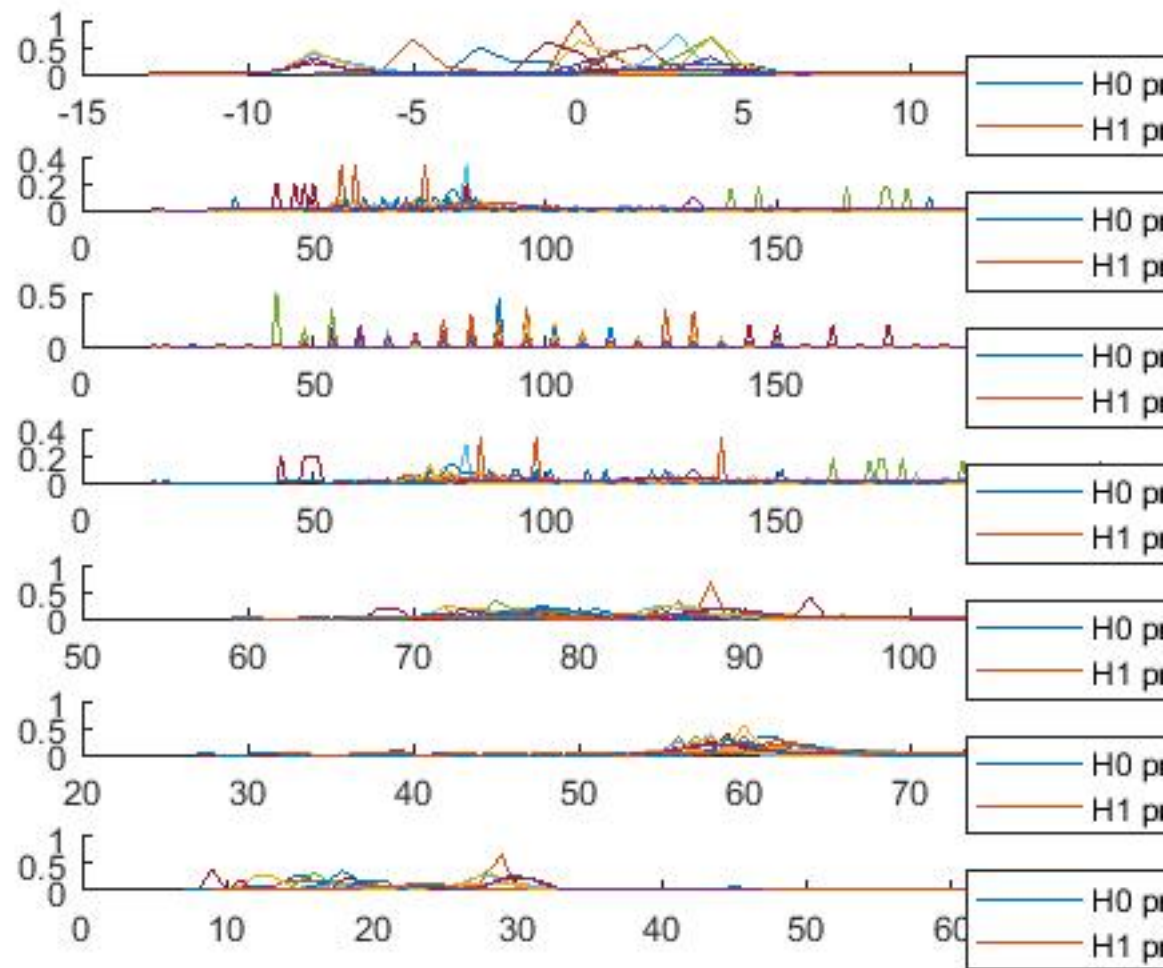
```

```

23     ML_vector(k) = 1;
24     end
25     if (P_H1*P_H1_i >= P_H0*P_H0_i)
26         % if H1_pmf*P(H1) >= H0_pmf*P(H0)
27         MAP_vector(k) = 1;
28     end
29 end

31 HT_table = cat(1, x_mat, feature_mat(1:1, :), feature_mat(2:2, :)
    , ML_vector, MAP_vector);
32 HT_table = rot90(HT_table, -1);
33 HT_table = fliplr(HT_table);
34 % Have to flip the table so that it matches with the given format
35 HT_table_array_pat{1, i} = num2cell(HT_table);
36 subplot(7, 1, i);
37 hold on;
38 plot(x_mat, feature_mat(1:1, :));
39 plot(x_mat, feature_mat(2:2, :));
40 legend('H0 pmf', 'H1 pmf');
41 hold off;
end

```



Task 1.2: