INDUSTRIALES
ETSII | UPM

CENTRE FOR AUTOMATION AND ROBOTICS

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Máster en Automática y Robótica

Applied Artificial Intelligence

Assignment 3.1: Cost Function

Josep María Barberá Civera (17048)

March 12, 2024

ASSIGNMENT 2.3

Cost Function

Main instructions for the task completion are presented here:

1. First load the data from "datos_D1_C2.mat" file.
2. Use `Bayesian classifier` for classifying input value 0.9, assuming that both classes have the same *a priori* probability.
3. Same as before, but assuming that the probability of every class is proportional to its number of instances in the original data.
4. Compute the class of less cost for the same input, assuming the following `Cost Matrix`:

$$\begin{array}{ll} C(c1/c1) = 1.1 & C(c1/c2) = 1.1 \\ C(c2/c1) = 8.0 & C(c2/c2) = 0.1 \end{array}$$

5. Discuss your results.

1.1 Methodology

In this case, the data to be used are of a different nature to those previously used. For example, they are one-dimensional as can be seen in Figure 1, the result of plotting them along the x-axis, setting the y-coordinate to zero.

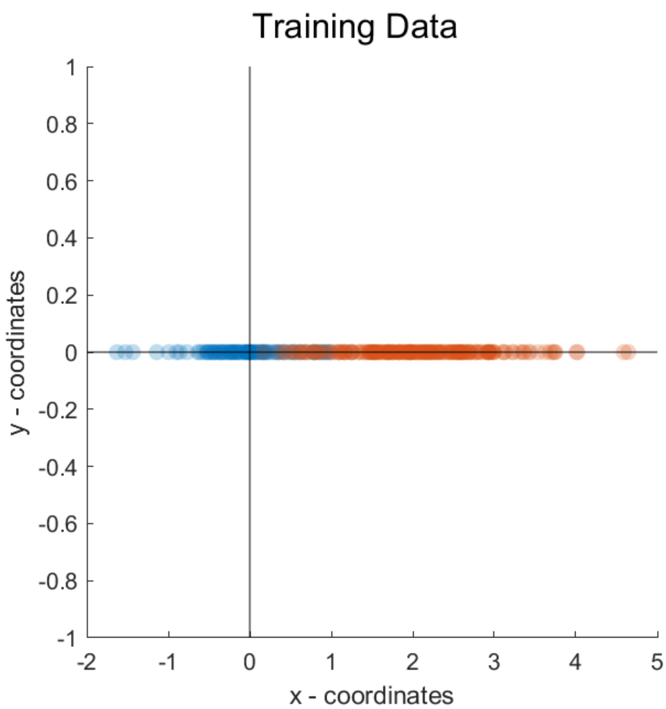

Figure 1: *Training data from "datos_D1_C2.mat" plotted using the colours convention from labels.*

Then again the `fitcnb` function has been used. First with a prior of [0.5 0.5] and then with a prior of [0.33 0.67]. Finally a prior value of [1.1 0.1] is introduced with a cross-cost matrix of [0 1.1; 8 0].

1.2 Discussion and Results

Once the model has been trained, each of the points in the domain is evaluated. In this way the probability distribution is obtained. The requested 0.9 point is located right on the boundary between zones. Thus in the first case with equal values of *a priori* probability the classification is `Class 1`. For this first case Figure 1 shows the values of the distribution.

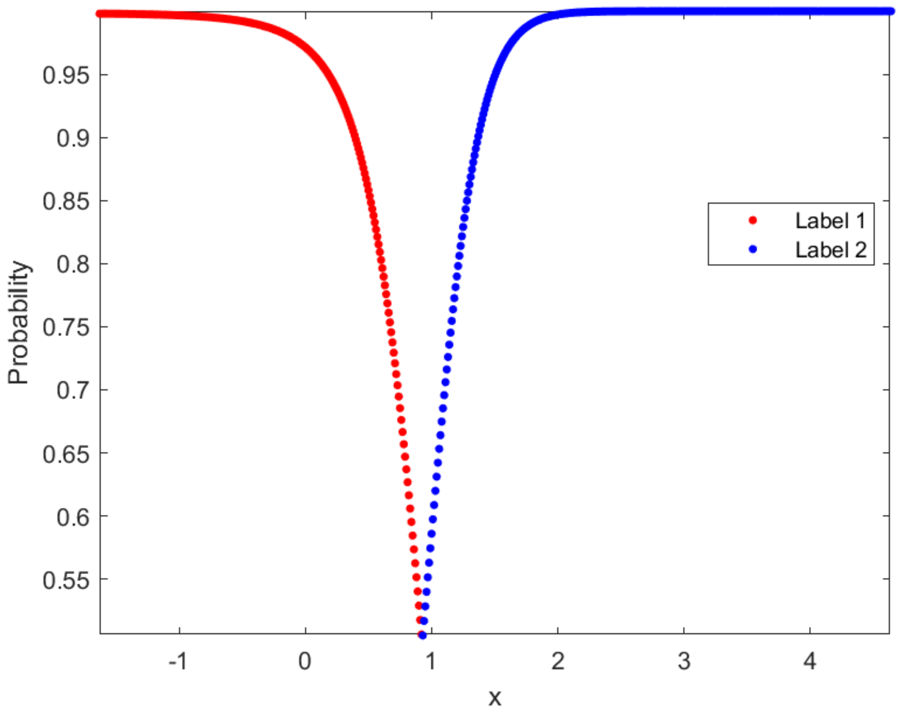

Figure 2

On the other hand, given different *a priori* probability values, the distribution changes slightly (as can be seen in Figure 3) leading to a classification value of `Class 2`.

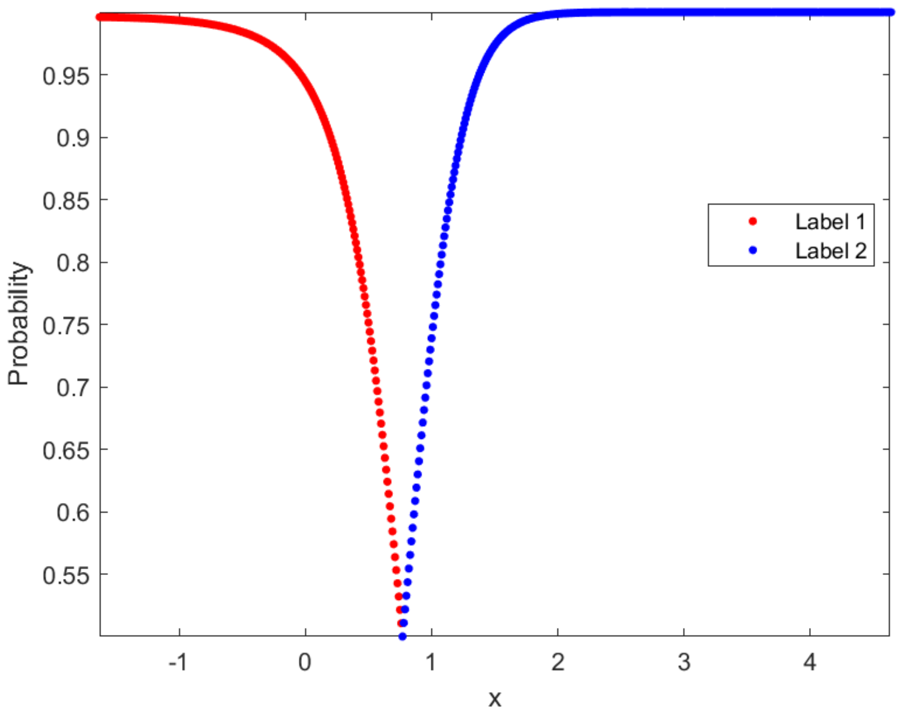

Figure 3

Finally for a given cost matrix the distribution takes discontinuous values (refer to Figure 4). And the classification remains `Class 2`.

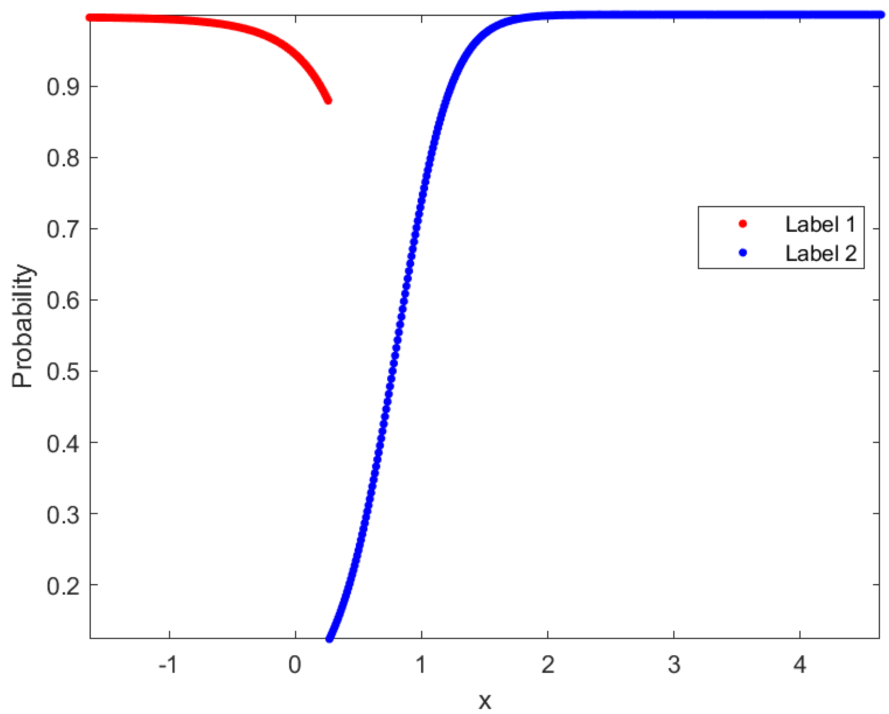

Figure 4

1.3 Relevant Code

The code prepared for this assignment is shown in the next pages in a wider style.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                           Master in Robotics
%                   Applied Artificial Intelligence
%
% Assinment 3.2:  Cost Function
% Student: Josep Barbera Civera
% ID: 17048
% Date: 12/03/2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 0. First load the data from ``datos\_D1\_C2.mat" file.
% 1. Use Bayesian classifier for classifying input value 0.9,
%     assuming that both classes have the same a priori probability.
% 2. Same as before, but assuming that the probability of every class
%     is proportional to its number of instances in the original data.
% 3. Compute the class of less cost for the same input, assuming the
%     following Cost Matrix:
%         C(c1/c1) = 1.1;   C(c1/c2) = 1.1
%         C(c2/c1) = 8.0;   C(c2/c2) = 0.1

load data_D1_C2.mat

%% Accesing Data
pvalues = p.value;
plabels = p.class;
tvalues = t.value;
tlabels = t.class;

[Mp, Np] = size(pvalues);
[Mt, Nt] = size(tvalues);

%% Plotting Normalized Data vs Raw Data
% one_plot('Training Data', 'x - coordinates', ...
%           'y - coordinates', 'Raw Data', pvalues, plabels, ...
%           'centred_data_vs_raw_data.png');

%% Models Trainning
% 1.
bayes_1 = fitcnb(pvalues', plabels', 'Prior', [0.5 0.5]);
% Model Prediction
[lab_1, Prob_1, Cost_1] = predict(bayes_1, 0.9);

% 2.
total = length(plabels);
label_1 = sum(plabels == 1);
label_2 = total - label_1;
prior = [label_1/total label_2/total];
bayes_2 = fitcnb(pvalues', plabels', 'Prior', prior);
% Model Prediction
[lab_2, Prob_2, Cost_2] = predict(bayes_2, 0.9);

% 3.
prior = [1.1 0.1];
```

```
cost = [0 1.1; 8 0];
bayes_3 = fitcnb(pvalues', plabels', 'Cost', cost);
% Model Prediction
[lab_3, Prob_3, Cost_3] = predict(bayes_3, 0.9);

%% pvalues prediction map vs predicted values
model = bayes_3;
X = min(pvalues(:)):0.01:max(pvalues(:));
[label,Prob,Cost] = predict(model,X');
figure
h1 = plot(X(label == 1), Prob(label == 1,1), 'r.', 'MarkerSize', 10);
hold on;
h2 = plot(X(label == 2), Prob(label == 2,2), 'b.', 'MarkerSize', 10);
xlabel('x')
ylabel('y')
title('{\bf knn classifier - tvalues (validation set)}')
legend([h1, h2], {'Label 1', 'Label 2'}, 'Location', 'best');
legend_box = findobj(gcf, 'Type', 'legend');
set(gcf, 'Color', 'white');
set(legend_box, 'Color', [0.8 0.8 0.8]); % Gray background color
axis tight
hold off
saveas(gcf, 'bayes_3.png'); % do not forget to change the name!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Auxiliar Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function one_plot(generic_title, ...
                    x_1_label, y_1_label, ...
                    legend_1, data_1, ...
                    labels_1, saved_name)
    disp("-------- Plotting ----------------------------------");
    disp(generic_title);
    figure;
    % Generic Title
    sgtitle(generic_title);
    % First Subplot
    subplot(1, 1, 1);
    for i=1:length(labels_1)
        if labels_1(i) == 1
            scatter(data_1(1,i), 0, 50, 'o', 'LineWidth', 10, ...
                    'MarkerEdgeColor', 'none', 'MarkerFaceColor', [0 0.4470 0.7410], ...
                    'MarkerFaceAlpha', 0.25); hold on;
        else
            scatter(data_1(1,i), 0, 50, 'o','LineWidth', 10, ...
                    'MarkerEdgeColor', 'none', 'MarkerFaceColor', [0.8500 0.3250 0.0980], ...
                    'MarkerFaceAlpha', 0.25); hold on;
        end
    end
    % Plot vertical lines for x=0 and y=0
    plot([0 0], ylim, 'k-');
    plot(xlim, [0 0], 'k-');
```

```
    % Subplot title
    % title(title_subplot_1);
    % Axis labels
    xlabel(x_1_label);
    ylabel(y_1_label);
    % legend({legend_1 "other legend"}, 'Location', 'best');
    pbaspect([1 1 1]);
    % pos = get(gcf, 'Position');
    % set(gcf, 'Position',pos+[-900 -300 900 300])
    saveas(gca, saved_name);
end
```