Assignment 1 Assignment 1

1

Using the code given in the appendix we created the scatterplot in figure 1.

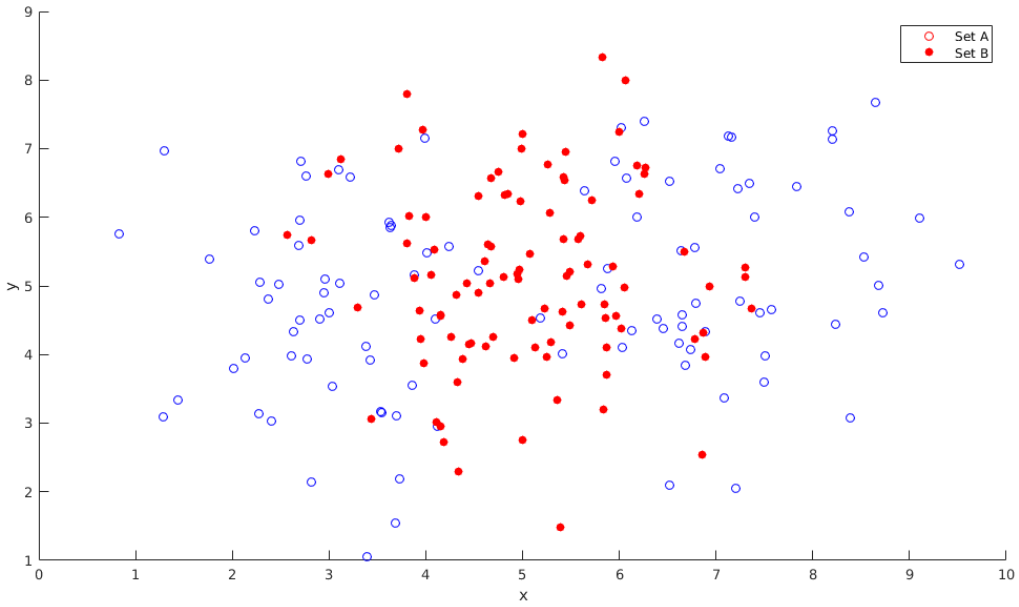

Figure 1: Scatterplot for the two classes.

The plot shows that there are at least three prototypes needed to approach a fairly well classification of these data. Two for set A, which should probably be located around $(3, 4.5)$ and $(7.5, 5.5)$, and one for set B somewhere around $(5, 5)$.

2

The code in the appendix shows our implementation of the LVQ1 algorithm. We acquired the following results for the different settings.

a 1 Prototype for class A and 1 prototype for class B

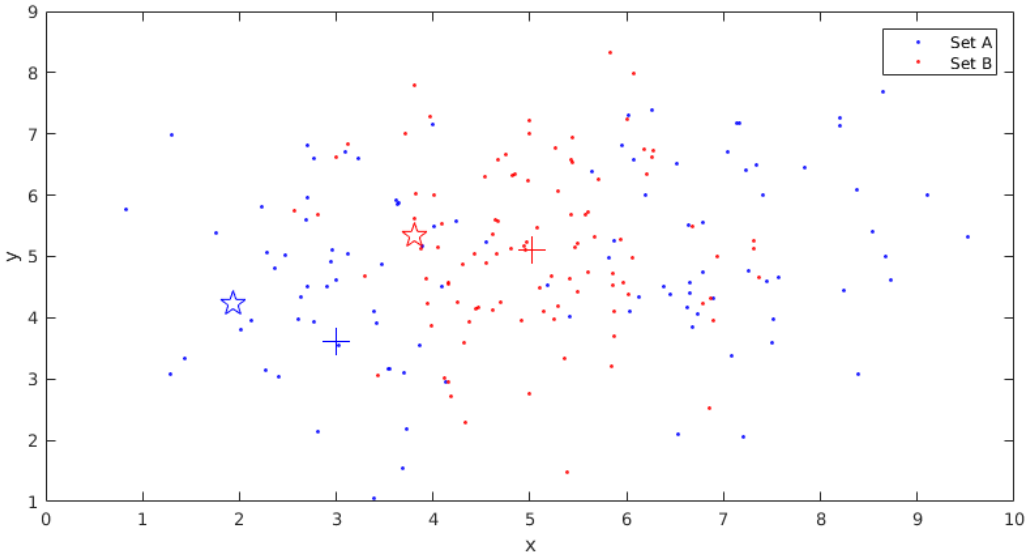

Figure 2: Scatterplot for the two classes.

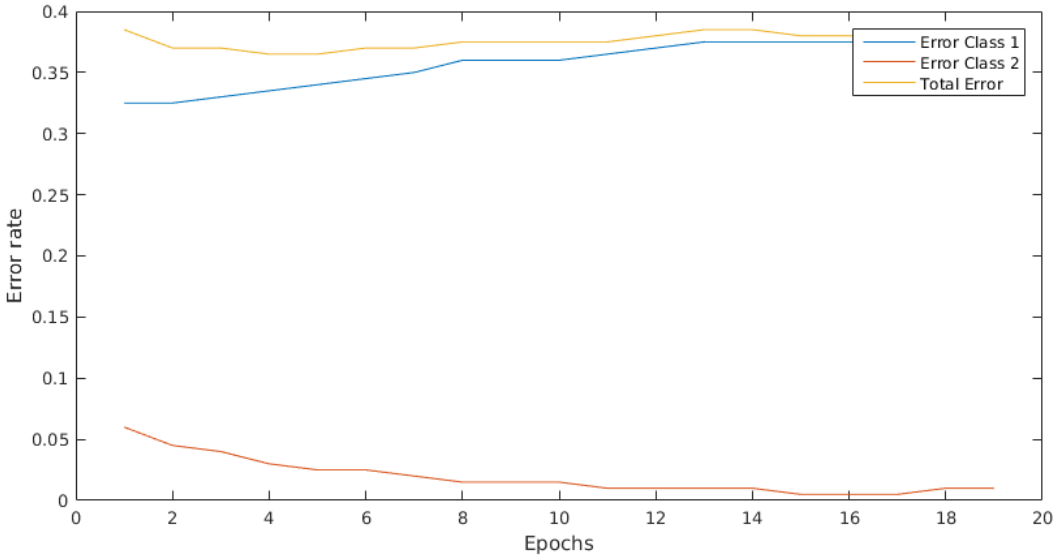

Figure 3: Training error rate to number of epochs.

As is expected with only one prototype per class, the prototype for class A is formed quite well, allowing it to correctly classify at least the data points that belong to class A. However, since class B is distributed in two groups, with A in between them, the prototype for class B is formed in the center of one of the two clusters, which means it can only correctly classify that cluster correctly. The other cluster will be incorrectly classified as class A.

b

Appendix

../Code/Ass1_1.m

```
hold off;
hold on;
scatter(matA(:,1),matA(:,2), 'blue');
scatter(matB(:,1),matB(:,2), 'red', 'filled');
xlabel('x'); ylabel('y'); legend('Set A', 'Set B');
```

../Code/Ass1_2.m

```
load('data_lvq_A') % matA
load('data_lvq_B') % matB

close all
figure;
plot(matA(:,1),matA(:,2), 'bp', 'markersize', 2);
hold on;
plot(matB(:,1),matB(:,2), 'rp', 'markersize', 2);
xlabel('x'); ylabel('y');
legend('Set A', 'Set B');

data = [matA ; matB];
data_labels = (floor( (0:length(data)-1) * 2 / length(data))).';
data = [data data_labels];

% The prototypes
w_A = 1;
w_B = 1;
w = zeros(w_A + w_B, ndims(data)+1);

eta = 0.01;
nrEpochs = 500;

E_1 = zeros(1,nrEpochs);
E_2 = zeros(1,nrEpochs);

% Randomly initialize the prototypes between the minimum and maximum values
% last value being their class
for i = 1 : size(w,1)
    if i <= w_A
        w(i,:) = [mean(matA) + rand()*2*std(matA)-std(matA)  0];
    else
        w(i,:) = [mean(matB) + rand()*2*std(matB)-std(matB)  1];
    end
end

plot(w(1:w_A,1), w(1:w_A,2), 'b+', 'markersize', 15);
plot(w(w_A+1:size(w,1),1), w(w_A+1:size(w,1),2), 'r+', 'markersize', 15);

for epoch = 1:nrEpochs
    % Training
    for point = 1 : size(data,1)
```

```
43            % Find the row with the nearest prototype
44            rowMin = find(pdist2(data(point,1:2), w(:,1:2)) == min(pdist2(data(point
                  ,1:2), w(:,1:2))),1);
45            % If the classes of the data point and the nearest prototype are the same
46            if w(rowMin,end) == data(point, end)
47                % Move the row closer to the data point
48                w(rowMin,1:2) = w(rowMin,1:2) + eta * (data(point,1:2) - w(rowMin,1:2));
49            else
50                w(rowMin,1:2) = w(rowMin,1:2) - eta * (data(point,1:2) - w(rowMin,1:2));
51            end
52        end
53
54        % Testing
55        for point = 1 : size(data,1)
56            % Find the row with the nearest prototype
57            rowMin = find(pdist2(data(point,1:2), w(:,1:2)) == min(pdist2(data(point
                  ,1:2), w(:,1:2))),1);
58            if w(rowMin,end) ~= data(point, end)
59                if point <= size(matA,1)
60                    E_1(epoch) = E_1(epoch) + 1;
61                else
62                    E_2(epoch) = E_2(epoch) + 1;
63                end
64            end
65        end
66        E = E_1 + E_2;
67
68        if (epoch > 10 && var(E(:,epoch-4:epoch)) < 0.05)
69            epoch
70            E_1(:,epoch+1:end) = [];
71            E_2(:,epoch+1:end) = [];
72            E(:,epoch+1:end) = [];
73            break
74        end
75    end
76
77    plot(w(1:w_A,1), w(1:w_A,2), 'bp', 'markersize', 15);
78    plot(w(w_A+1:size(w,1),1), w(w_A+1:size(w,1),2), 'rp', 'markersize', 15);
79
80    figure;
81    plot(E_1/200)
82    hold on;
83    plot(E_2/200);
84    plot(E/200);
85    legend('Training error Class 1', 'Training error Class 2', 'Total training Error');
86    xlabel('Epochs')
87    ylabel('Error rate')
```

Pattern Recognition Practical 4

Group 24: Maikel Withagen (s1867733) Steven Bosch (s1861948)

October 8, 2015