Introduction to Intelligent Systems - Lab 4

Diego Velasco Volkmann Eduardo Faccin Vernier
S2851059 S3012875

October 31, 2015

Assignment 1:

Normal distributions Consider the following two sets of observed 1D data (feature values) coming from objects of two different classes:
S1 = [-30.8508 23.6509 -15.4613 -0.0466 26.5835 41.1727
-23.5292 47.8198 29.8043 7.9640 4.1434 3.4718
0.9068 10.6914 11.5387 34.7819 55.8093 24.0074
3.7086 28.7557 15.4968 -20.8930 38.4767 19.2119
14.0552 25.4574 17.8422 -18.2446 5.1299 5.6184]

S2 = [44.6799 66.8210 41.2427 45.1618 42.8800 38.2579
48.0776 47.2593 65.3007 47.5098 39.3579 66.0346
62.3468 47.7037 34.9384]

and the following test set of data points that need to be classified:
T = [5 23 40 70 95].

- Plot the elements of the two sets S1 and S2 on the x-axis of a Fig.1: the elements of S1 as blue circles (bo) and the elements of the set S2 as red circles (ro). Plot in the same Fig.1 the points of the test set T as black squares (ks).
  Answer: Figure 1 shows the plot resulting from the Listing 1 code's execution, which simply plots the matrixes in the requested way.

Listing 1: Part of file assign1.m used to plot S1, S2 and T

```
load('X:\My Desktop\normdist(1).mat')
figure;
hold on;
plot(S1,0,'bo');
plot(S2,0,'ro');
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('S1, S2 and T plot');
xlabel('Point values');
```

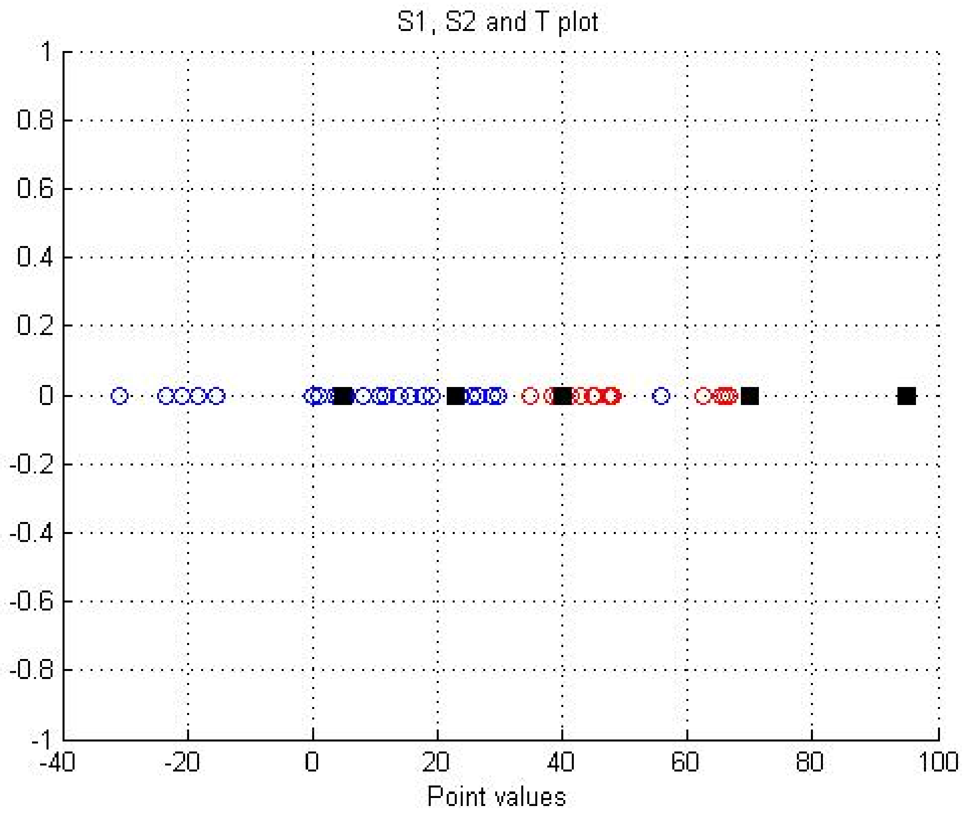

Figure 1: Plot of S1(blue circles), S2(red circles) and T(black squares)

- After a brief consideration of what the underlying distributions that produced the training data sets S1 and S2 might be, decide that they are normal distributions. Compute the parameters (mean and standard deviation) of the two distributions, using maximum estimation. Create a Fig.2 in which you plot the two Gaussian functions, one in blue the other in red, together with the points of the two training data sets S1 and S2. For a moment, contemplate on the elegance of the functions in comparison to the rugged data in the background. The two functions are the class conditional probability densities p(x—1) and p(x—2) that the two classes produce a value x of the considered feature.
  Answer: Figure 2 shows the resulting plot for Listing 2 code's execution, which uses the function "mean(matrix(:))", to calculate the mean of the whole matrix, and "std(matrix(:))", to calculate the standard deviation of the whole matrix, to get the gaussian function by using the function "gaussmf".

Listing 2: Part of file assign1.m used to plot the Gaussian function of S1 and S2

```
load('X:\My Desktop\normdist(1).mat')
figure;
hold on;
meanS1 = mean(S1(:));
stdS1 = std(S1(:));
meanS2 = mean(S2(:));
stdS2 = std(S2(:));
meanT = mean(T(:));
stdT = std(T(:));
range = -60:0.1:100;
plot(range,gaussmf(range,[stdS1 meanS1]),'b');
plot(range,gaussmf(range,[stdS2 meanS2]),'r');
plot(S1,0,'bo');
plot(S2,0,'ro');
grid on;
title('S1 and S2 Gaussian functions');
xlabel('Point values');
ylabel('Probability');
```

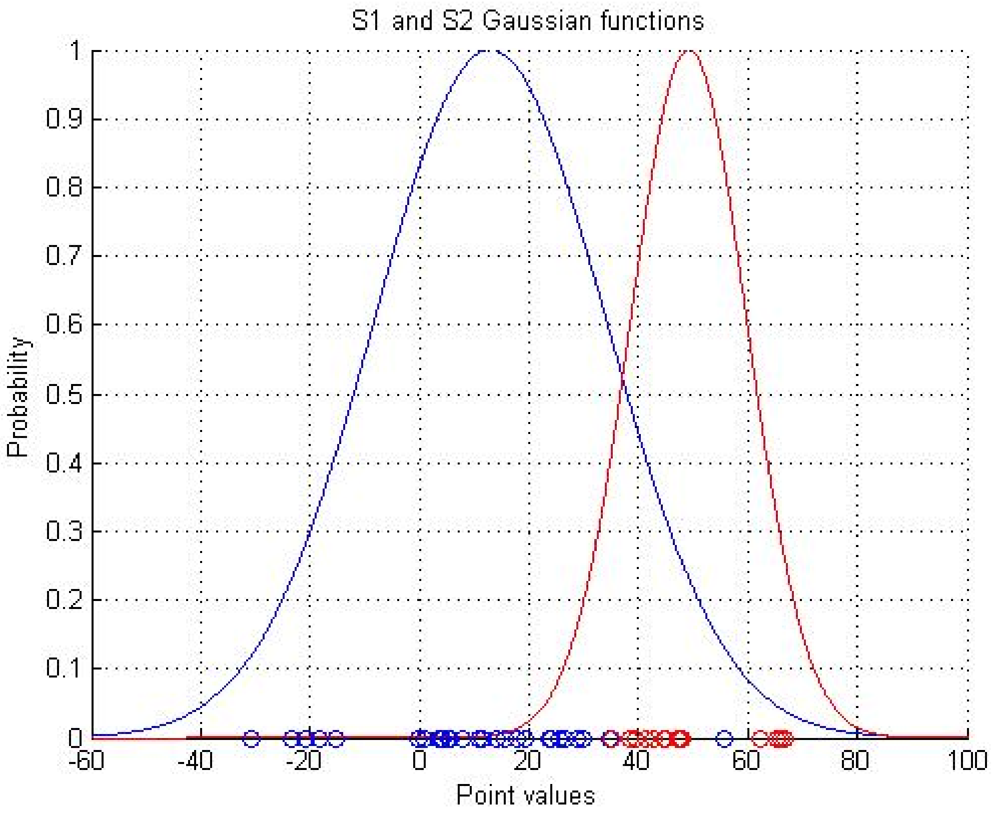

Figure 2: Plot of S1(blue) and S2(red) Gaussian functions

- Estimate the prior probabilities $P(\omega 1)$ and $P(\omega 2)$. If you have no clue how to do that, look at the lecture slides and follow the steps of the example given there.
  Answer: $P(\omega 1)$ = 0.6667 and $P(\omega 2)$ = 0.3333 were the results obatined by executing the code on Listing 3, which executes the following algorithm $P(\omega 1)$ = card(S1)/(card(S1)+card(S2)) and $P(\omega 2)$ = card(S2)/(card(S1)+card(S2)), but in MatLab the function "card" is called "numel".

Listing 3: Part of file assign1.m used to calculate $P(\omega 1)$ and $P(\omega 2)$

```
load('X:\My Desktop\normdist(1).mat')
pw1=numel(S1)/(numel(S1)+numel(S2));
pw2=numel(S2)/(numel(S1)+numel(S2));
```

- Create a Fig.3 in which you plot the two products $P(\omega 1)p(x|\omega 1)$ and $P(\omega 2)p(x|\omega 2)$, together with the points of the two training data sets S1 and S2 and the test set T.
  Answer: Figure 3 shows the resulting plot for Listing 4 code's execution, which simply take the result from the previous item and uses it to multiply the gaussian function from item 2.

Listing 4: Part of file assign1.m used to plot $P(\omega 1)p(x|\omega 1)$ and $P(\omega 2)p(x|\omega 2)$

```
load('X:\My Desktop\labelsAEX.mat')
load('X:\My Desktop\normdist(1).mat')
load('X:\My Desktop\dataAEX.mat')
pw1=numel(S1)/(numel(S1)+numel(S2));
pw2=numel(S2)/(numel(S1)+numel(S2));
figure;
hold on;
meanS1 = mean(S1(:));
stdS1 = std(S1(:));
meanS2 = mean(S2(:));
stdS2 = std(S2(:));
meanT = mean(T(:));
stdT = std(T(:));
range = -60:0.1:100;
plot(range,(((pw1*gaussmf(range,[stdS1 meanS1]))-min(pw1*gaussmf(
    range,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))
    +min(pw1*gaussmf(range,[stdS1 meanS1])))),'b');
plot(range,(((pw2*gaussmf(range,[stdS2 meanS2]))-min(pw1*gaussmf(
    range,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))
    +min(pw1*gaussmf(range,[stdS1 meanS1])))),'r');
plot(S1,0,'bo');
plot(S2,0,'ro');
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('P(w1)p(x|w1) and P(w2)p(x|w2), where "w" is omega');
xlabel('Point values');
ylabel('Probability');
```

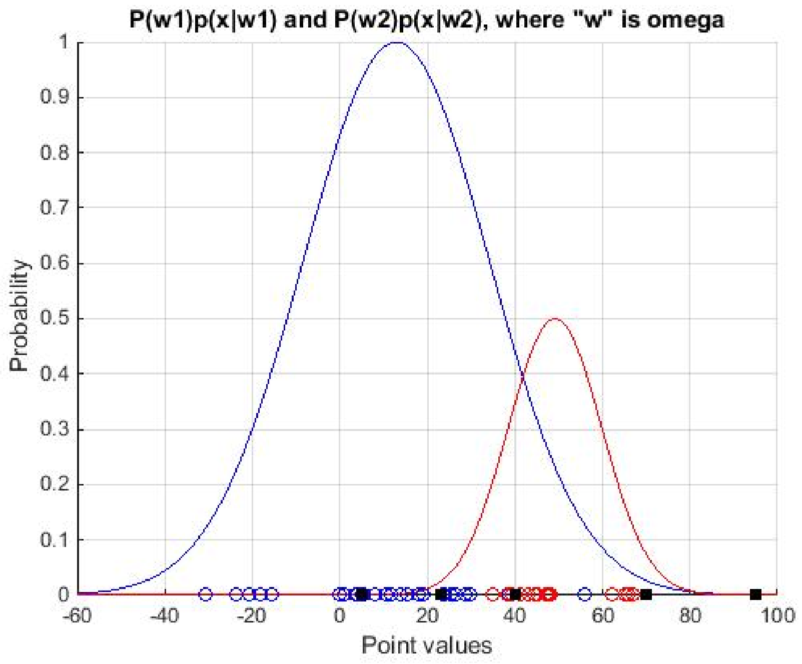

Figure 3: Plot of S1(blue) and S2(red) Gaussian functions multiplied by their respective prior probability

- Substitute the values of the prior probabilities and the class conditional probability densities determined above in the equation $P(\omega 1)p(x|\omega 1) = P(\omega 2)p(x|\omega 2)$ and solve the resulting equation for x in order to determine the value(s) of the decision criterion that we should use for classification.
  Answer:

$$P(\omega 1) * p(x|\omega 1) \; = \; P(\omega 2) * p(x|\omega 2)$$

$$0.6667 * p(x|\omega 1) \; = \; 0.3333 * p(x|\omega 2)$$

$$0.6667*(1/\sigma_1*sqrt(2\pi))*exp(-(x-\mu_1)^2/2*\sigma_1^2) \; = \; 0.3333*(1/\sigma_2*sqrt(2\pi))*exp(-(x-\mu_2)^2/2*\sigma_2^2)$$

$$0.6667 * (1/21.2131 * sqrt(2\pi)) * exp(-(x - 12.9024)^2/2 * 21.2131^2) \; =$$
$$0.3333 * (1/10.6773 * sqrt(2\pi)) * exp(-(x - 49.1715)^2/2 * 10.6773^2)$$

$$0.6667 * (0.118164) * exp(-(x - 12.9024)^2/899.99122322) \; =$$
$$0.3333 * (0.234762) * exp(-(x - 49.1715)^2/228.00947058)$$

$$0.0787799388 * exp(-(x - 12.9024)^2/899.99122322) \; =$$
$$0.0782461746 * exp(-(x - 49.1715)^2/228.00947058)$$

$$(0.0787799388/0.0782461746) * exp(-(x - 12.9024)^2/899.99122322) \; =$$
$$exp(-(x - 49.1715)^2/228.00947058)$$

$$1.0068216012185725434812502642141945684332534769054383*exp(-(x-12.9024)^2/899.99122322) \; =$$
$$exp(-(x - 49.1715)^2/228.00947058)$$

$$1.0068216012185725434812502642141945684332534769054383*exp(-(166.472-25.8048*x+x^2)/$$
$$899.99122322) \; = exp(-(2417.84 - 98.343 * x + x^2)/228.00947058)$$

$$ln(1.0068216012185725434812502642141945684332534769054383)-(166.472-25.8048*x+x^2)/$$
$$899.99122322 \; = ln1 - (2417.84 - 98.343 * x + x^2)/228.00947058$$

$$0.0067984393712-(166.472-25.8048*x+x^2)/899.99122322) \; = 0-(2417.84-98.343*x+x^2)/228.00947058$$

$$(166.472-25.8048*x+x^2)/899.99122322-0.0067984393712 \; = (2417.84-98.343*x+x^2)/228.00947058$$

$$(166.472 - 25.8048 * x + x^2) - (899.99122322 * 0.0067984393712) \; =$$
$$(2417.84 - 98.343 * x + x^2) * (899.99122322/228.00947058)$$

$$(166.472 - 25.8048 * x + x^2) - 6.118535765673295639264 \; =$$
$$(2417.84 - 98.343 * x + x^2) * 3.9471659704776461132204958606855776683414287676821989$$

$$x*(-25.8048+x) \; = x*(-388.176+3.94717*x)+9543.62-166.472+6.118535765673295639264$$

$$x * (-25.8048 + x) \; = x * (-388.176 + 3.94717 * x) + 9383.266535765673295639264$$

$$x * (-25.8048 + x) - (x * (-388.176 + 3.94717 * x) + 9383.266535765673295639264) \; = 0$$

$$x * (-25.8048 + x) - (x * (-388.176 + 3.94717 * x)) \; = 9383.266535765673295639264$$

$$(-25.8048 * x + x^2) - (-388.176 * x + 3.94717 * x^2) \; = 9383.266535765673295639264$$

$$362.371 * x - 2.94717 * x^2 \; = 9383.266535765673295639264$$

$$-362.371 * x + 2.94717 * x^2 + 9383.266535765673295639264 \; = 0$$

$$x = (362.371 \pm sqrt((362.371)^2 - 4*(2.94717)*(9383.266535765673295639264)))/2*(2.94717)$$

$$x = (362.371 \pm sqrt((131312.741641) - (110616.3265448500774668366787315 2)))/5.89434$$

$$x = (362.371 \pm sqrt(20696.415096149922533163321268 48))/5.89434$$

$$x = (362.371 \pm 143.862486757840810421003802648 4)/5.89434$$

$$x_1 = (362.371 + 143.86248675784081042100380264 84)/5.89434$$

$$x_1 = 85.88467695413579983865942627137219773545469043 1838$$

$$x_2 = (362.371 - 143.86248675784081042100380264 84)/5.89434$$

$$x_2 = 37.07090416266438474519559396838322865664349189 2222029$$

Therefore, the values of the decision criterion that should be used for classification are $\simeq 85.88468$ and $\simeq 37.07090$.

- Using the thus obtained value(s) of the decision criterion, determine the classes of the points in the test set [5 23 40 70 95]. Create a Fig.4 which copies Fig.3 and adds to it a specification of which domain on the x-axis belongs to class 1 (blue) and which complementary domain corresponds to class 2 (red).
  Answer: Figure 4 shows the resulting plot for Listing 5 code's execution, which simply take the result from the previous item and adds colored squares on the x-axis to represent the point's domain.

Listing 5: Part of file assign1.m used to plot the domain of points on the X-axis

```
load('X:\My Desktop\normdist(1).mat')
pw1=numel(S1)/(numel(S1)+numel(S2));
pw2=numel(S2)/(numel(S1)+numel(S2));
x1 = 37.07090;
x2 = 85.88468;
figure;
hold on;
meanS1 = mean(S1(:));
stdS1 = std(S1(:));
meanS2 = mean(S2(:));
stdS2 = std(S2(:));
meanT = mean(T(:));
stdT = std(T(:));
range = -60:0.1:100;
plot(range,(((pw1*gaussmf(range,[stdS1 meanS1]))-min(pw1*gaussmf(
    range,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))
    +min(pw1*gaussmf(range,[stdS1 meanS1])))),'b');
plot(range,(((pw2*gaussmf(range,[stdS2 meanS2]))-min(pw1*gaussmf(
    range,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))
    +min(pw1*gaussmf(range,[stdS1 meanS1])))),'r');
    for n=-60:0.1:x1
        plot(n,0,'bs','MarkerFaceColor','b');
    end
    for o=x1:0.1:x2
        plot(o,0,'rs','MarkerFaceColor','r');
    end
    for p=x2:0.1:100
        plot(p,0,'bs','MarkerFaceColor','b');
    end
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('X-axis domain');
xlabel('Point values');
ylabel('Probability')
```

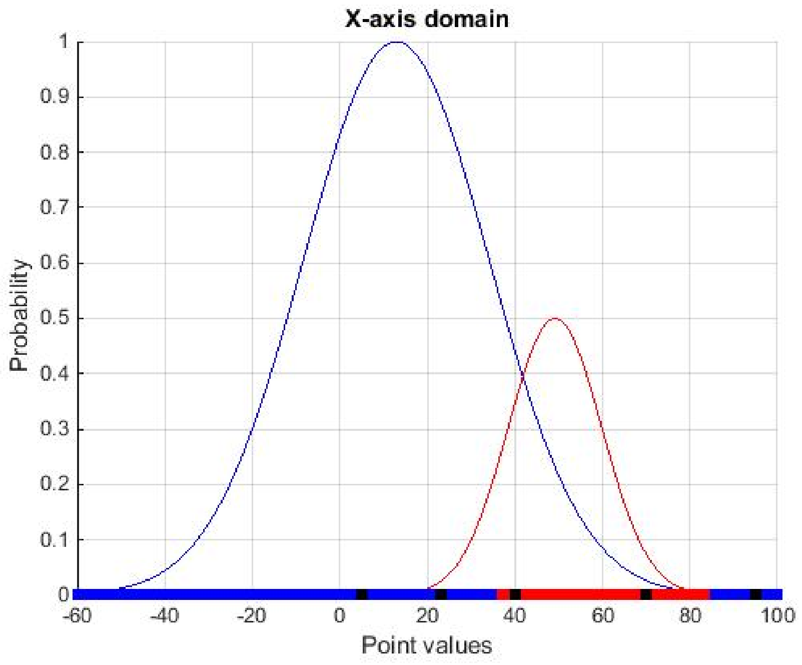

Figure 4: Plot of S1(blue) and S2(red) Gaussian functions multiplied by their respective prior probability with a colored line at the x-axis showing the estimated domain for the classes

- Evaluate the misclassification rate of the thus created classifier for each of the two classes. (For this, you can for instance use the error function erf, familiar from the iris recognition assignment in the first week.)
  Answer: The classifier found misclassificates 2 out of the 3 points in class T, which means there's a 40% misclassification rate in this case. However, if trying to classify the following array of 16 numbers "-60,-50,-40,-20,-10,0,10,20,30,40,50,60,70,80,90,100", it results in 3 misclassifications, meaning 18% error rate. The main problem is that any number beyond 85.88468 is classified as a class 1, when it should actually be a class 2.

Assignment 2:

The file dataAEX.mat contains 19 time series of 19 stocks whose names are specified in labelsAEX.mat. You can load the data of these files in your program using importdata or load. Create a dendrogram of the 19 stocks using the Matlab functions linkage and dendrogram. Apply the complete linkage algorithm, using Euclidean distance between the time series.

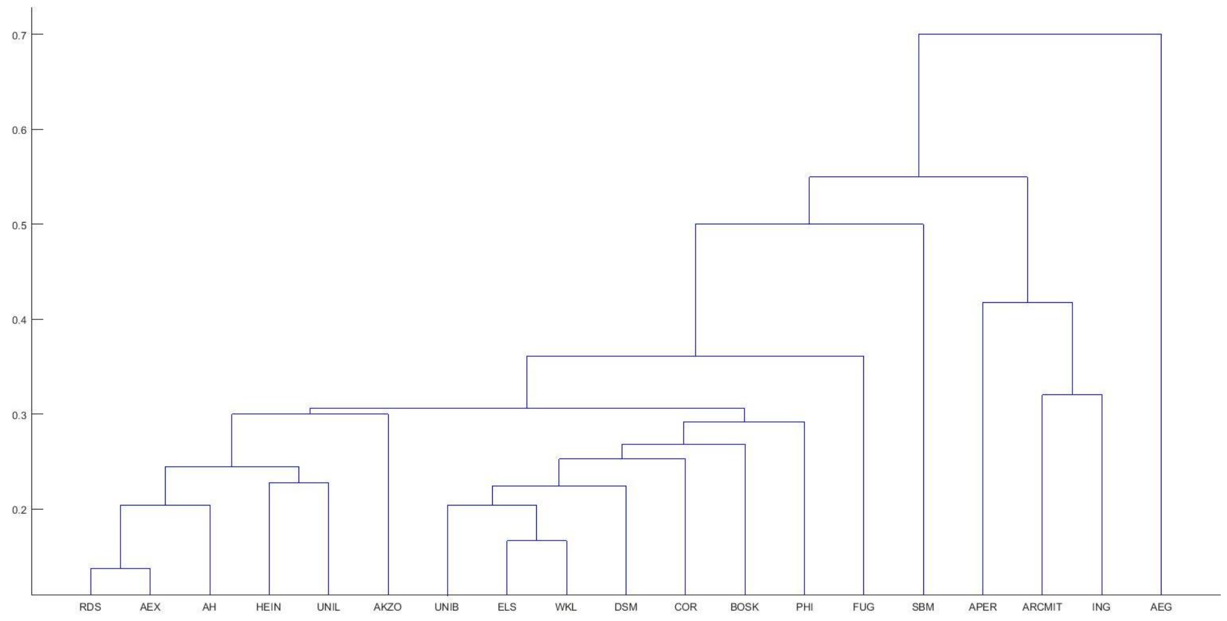

Figure 5: Dendrogram of the dissimilarity in stock behaviour

Listing 6: dend.m file

```
dataAEX = load('dataAEX.mat');
labelsAEX = load('labelsAEX.mat');

dataAEX = dataAEX.data;
L = linkage(dataAEX, 'complete');
figure;
[h, nodes, origin] = dendrogram(L);
hAxis = get(h(1), 'parent')
perm = str2num(get(hAxis, 'XtickLabel'));
set(hAxis, 'XtickLabel', labels(perm));
```

In this assignment the linkage function was used to create an agglomerative hierarchical cluster three using euclidean distance as the metric and the 'complete' method which uses the further distance to compute distance between clusters. The dendrogram was generated using matlab's dendrogram function.
This technique might be used in the stock market to benefit investors as a way to know how to choose stocks that are not very correlated to each other. This way, it is possible to choose a set of stocks that in situations where one stock goes down, the rest the set doesn't folow the same trend, therefore providing more security to the investor's money.

Division of work

Diego was responsible for assignment 1, while Eduardo was responsible for assignment 2.

Appendix:

Listing 7: assign1.m file

```
load('X:\My Desktop\labelsAEX.mat')
load('X:\My Desktop\normdist(1).mat')
load('X:\My Desktop\dataAEX.mat')
figure;
hold on;
plot(S1,0,'bo');
plot(S2,0,'ro');
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('S1, S2 and T plot');
xlabel('Point values');

figure;
hold on;
meanS1 = mean(S1(:));
stdS1 = std(S1(:));
meanS2 = mean(S2(:));
stdS2 = std(S2(:));
meanT = mean(T(:));
stdT = std(T(:));
%fx = sort(S1);
%fx2 = sort(S2);
%fx = (fx-min(fx))/(max(fx)-min(fx));
%pd = makedist('Normal',meanS1,stdS1);
%pd2 = makedist('Normal',meanS2,stdS2);
range = -60:0.1:100;
plot(range,gaussmf(range,[stdS1 meanS1]),'b');
plot(range,gaussmf(range,[stdS2 meanS2]),'r');
plot(S1,0,'bo');
plot(S2,0,'ro');
%plot(fx,pdf(pd,fx),'b');
%plot(fx2,pdf(pd2,fx2),'r');
grid on;
title('S1 and S2 Gaussian functions');
xlabel('Point values');
ylabel('Probability');

pw1=numel(S1)/(numel(S1)+numel(S2));
pw2=numel(S2)/(numel(S1)+numel(S2));

figure;
hold on;
%plot(range,(pw1*gaussmf(range,[stdS1 meanS1])),'b');
%plot(range,(pw2*gaussmf(range,[stdS2 meanS2])),'r');
plot(range,(((pw1*gaussmf(range,[stdS1 meanS1]))-min(pw1*gaussmf(range
    ,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))+min(pw1*
    gaussmf(range,[stdS1 meanS1])))),'b');
plot(range,(((pw2*gaussmf(range,[stdS2 meanS2]))-min(pw1*gaussmf(range
    ,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))+min(pw1*
    gaussmf(range,[stdS1 meanS1])))),'r');
```

```
plot(S1,0,'bo');
plot(S2,0,'ro');
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('P(w1)p(x|w1) and P(w2)p(x|w2), where "w" is omega');
xlabel('Point values');
ylabel('Probability');

x1 = 37.07090;
x2 = 85.88468;

figure;
hold on;
plot(range,(((pw1*gaussmf(range,[stdS1 meanS1]))-min(pw1*gaussmf(range
    ,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))+min(pw1*
    gaussmf(range,[stdS1 meanS1])))),'b');
plot(range,(((pw2*gaussmf(range,[stdS2 meanS2]))-min(pw1*gaussmf(range
    ,[stdS1 meanS1])))/(max(pw1*gaussmf(range,[stdS1 meanS1]))+min(pw1*
    gaussmf(range,[stdS1 meanS1])))),'r');
    for n=-60:0.1:x1
        plot(n,0,'bs','MarkerFaceColor','b');
    end
    for o=x1:0.1:x2
        plot(o,0,'rs','MarkerFaceColor','r');
    end
    for p=x2:0.1:100
        plot(p,0,'bs','MarkerFaceColor','b');
    end
plot(T,0,'ks','MarkerFaceColor','k');
grid on;
title('X-axis domain');
xlabel('Point values');
ylabel('Probability');
```