CS 584 - HM4

Alaa Ayach A20317680

Problem Statement

In this assignment I am trying to implement Support Vector Machines for both hard and soft margins. I analyzed the differences comparing the different results based on confusion Matrix. Dataset was generated randomly.

Proposed Solution

I implemented all functions from scratch using only matrix manipulation given by R. I used aggregate function to compute sums.
I used confusion Matrix to come up with the error.

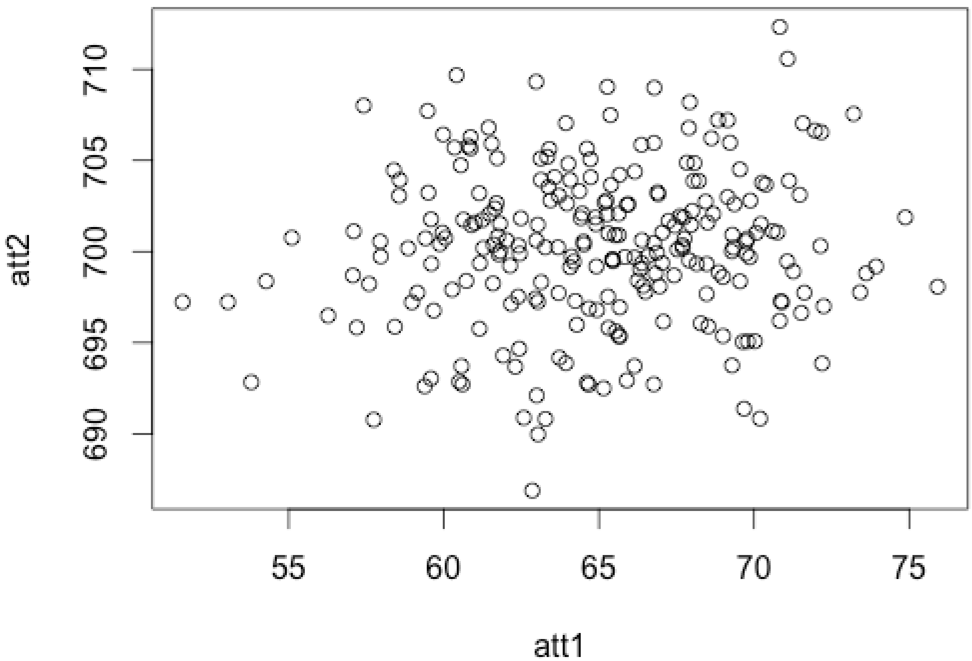
Implementation Details

I encapsulate all the work in a modular way using files and parameters. Those files took the parameters that I talked about above. used some package like cvTools to help in cross validation.
There were some problem in computing G matrix so I used gausskernel function provided by the package "KRLS" which only can generate G for pair-wise Gaussian matrix by passing features array and Sigma.

Results and Discussion

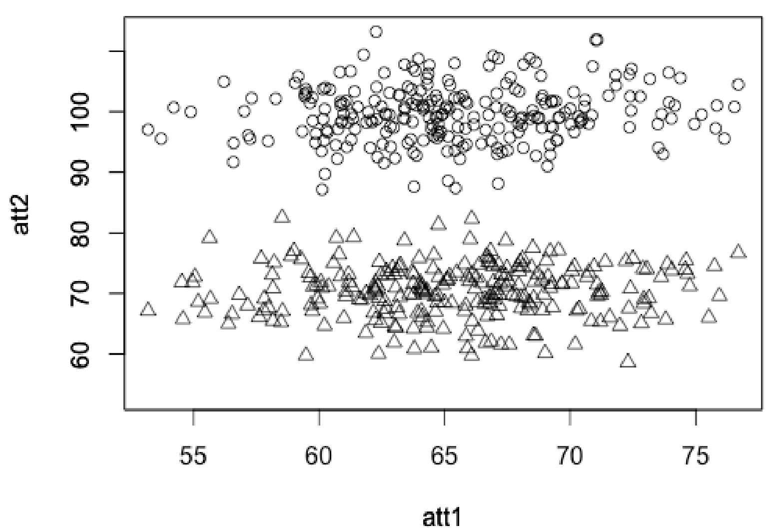
1. Generating Datasets

I used the function *rnorm()* that generates a dataset of points in the normal distribution. I combined two attributes in one dataframe and plot it as shown below:

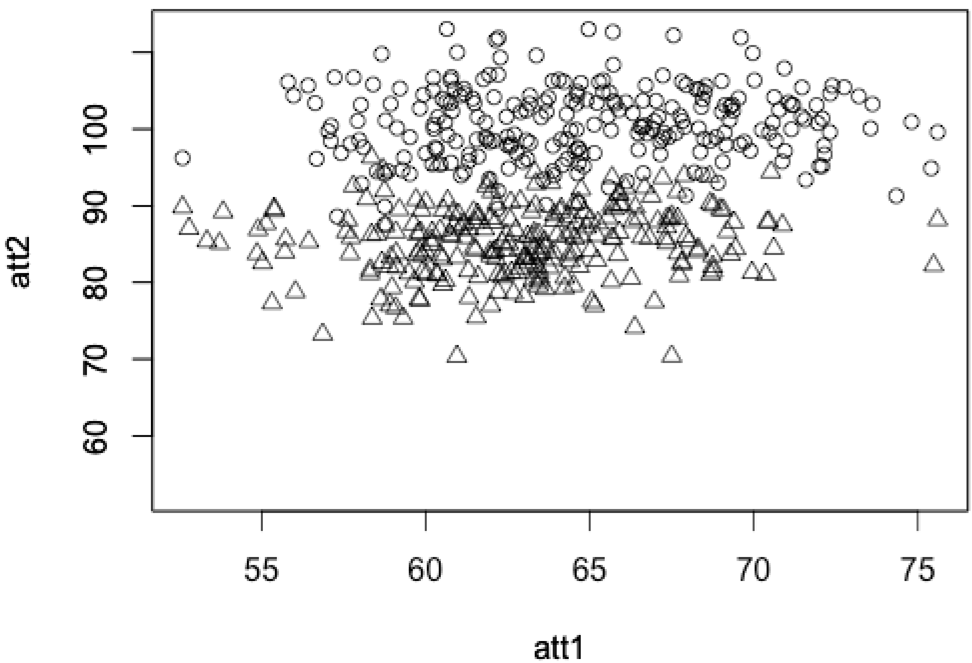


I made two cases for two datasets, the **first one was separate** as the means for the two classes differ in a great deal, however the **second one was not separate** so the means were very close.

Separable Dataset



Non-Separable Dataset



2. Hard margins SVM (Hard margins SVM.R)

I applied hard margins SVM using *ipop()* function from the package kernlab on a small dataset
This is the confusion matrix that I got:

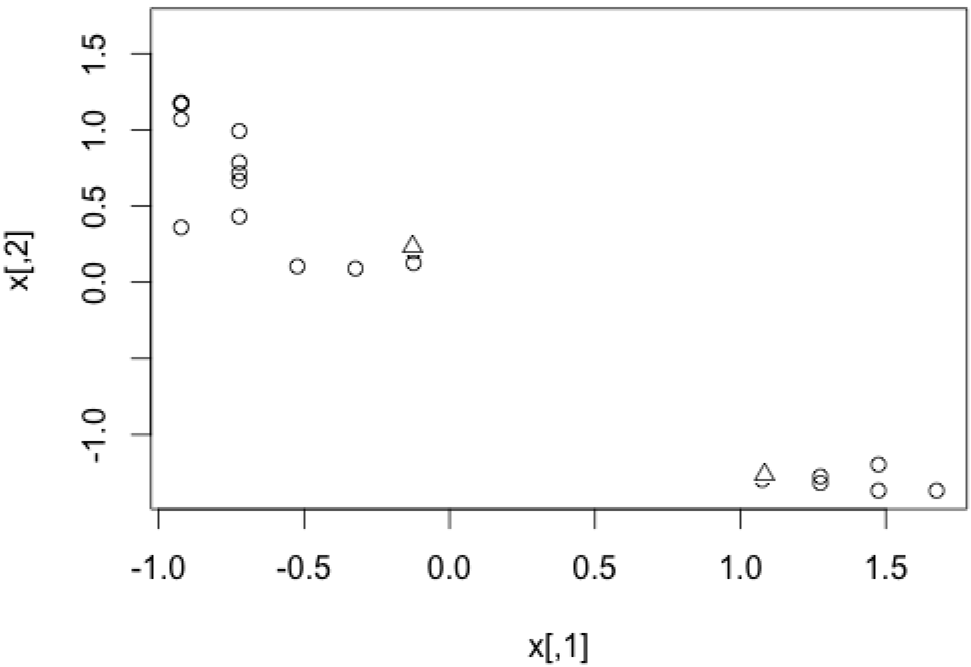
Confusion matrix:

y

yhat	-1	1
-1	14	0
1	0	7

<u>The accuracy is %100 as the data is separable.</u>

This graph shows the support vectors in triangles where SVMs are the points that have alpha close to 1



3. Developing the expression of the dual LD

$\partial Lp / \partial w0 = 0$

- Σ αi yi = 0

Σ αi yi = 0

$\partial Lp / \partial w = 0$

½ * 2 * w - Σ αi xi yi = 0

w = Σ αi xi yi

$\partial Lp / \partial \zeta = 0$

c αi - ***β*** = 0

So while computing Ld we omit the negative-positive contradictory parts and we will be left with:

Ld = ½ Σ Σ αi αj xi xj yi yj

S.T. αi > 0,

Σ αi yi =0

βi>0

c- αi - ***β***i = 0

0< αi < c

4. Soft margins SVM (Soft margins SVM.R)

I applied soft margins using c = 0.01 and got this confusion matrix:

```
    y
yhat -1  1
  -1 14  5
  1   0  2
```

Here we saw that many points become part for the first class and we have a great error here
The Support vectors are still the same in this case

5. Kernel (kernel.R)

I used gausskernel() function which gives Gaussian pair-wise matrix and I got this confusion matrix

```
    y
yhat -1  1
  -1 14  0
  1   0  7
```

It is %100 accuracy as I used the hard margins as well

6. Testing

When adding more points to the majority class (as we saw in the Soft Margins example), the points from the minority class were misclassified and assumed to belong to the majority class.
This problem will be more clear when adding more samples to the majority class

7. Improving suggestion

An initial suggestion would be to use a good split of the data for the training set, in a way that we should consider that the number of points from both classes is nearly identical

In complicated cases, choosing the C limit for better soft margining is a big deal so we can here do some optimizations on this value