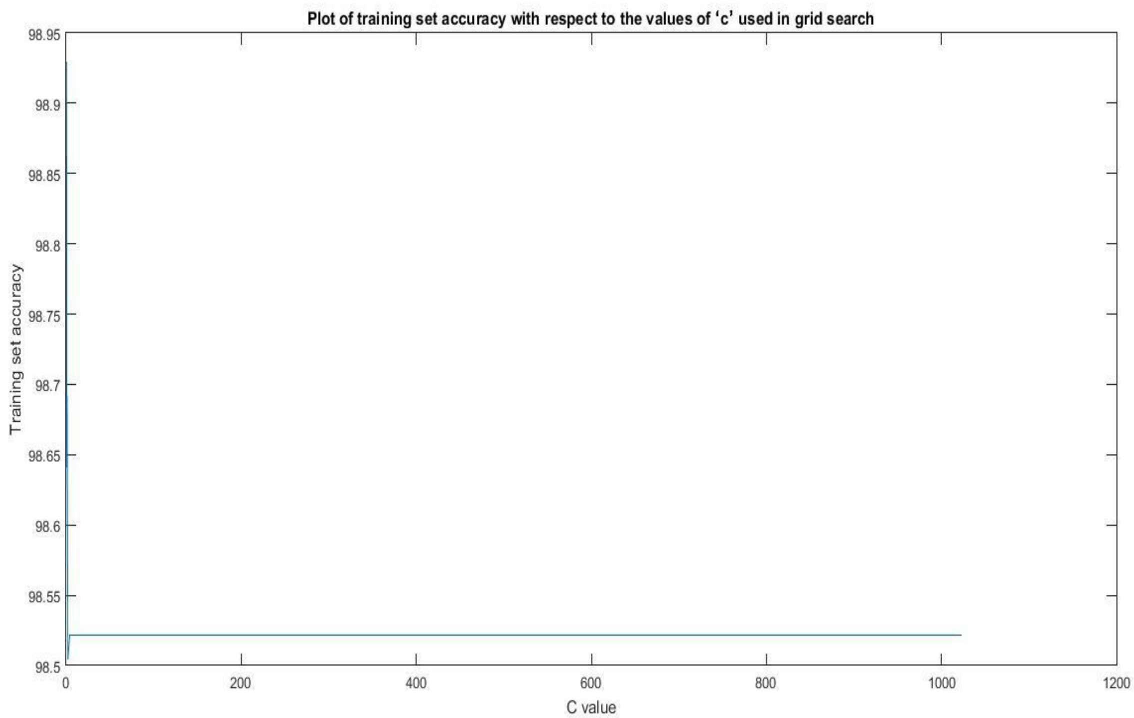
Q1(a) **Performing a grid search for the best value of ‘c’ :**

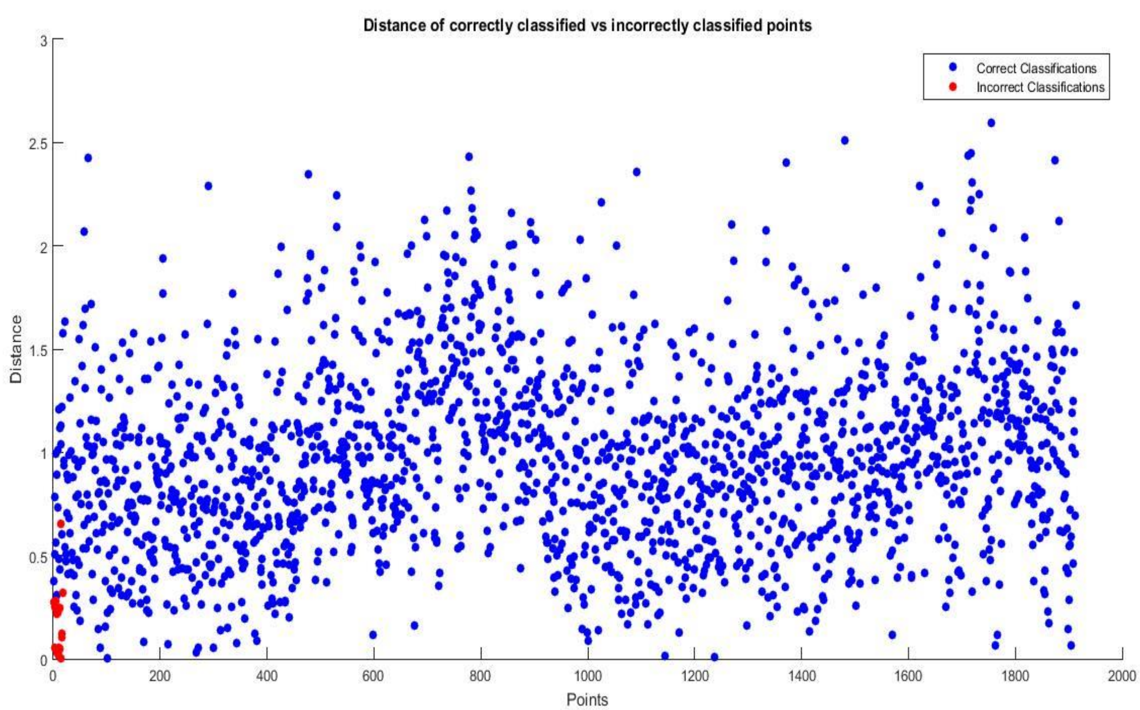
.........*......*
optimization finished, #iter = 15220
nu = 0.000126
obj = -47.412747, rho = -0.706443
nSV = 222, nBSV = 0
Total nSV = 222
Accuracy = 98.5217% (5798/5885) (classification)
7 98.5217 (best c=0.25, rate=98.9295)
.........*......*
optimization finished, #iter = 15220
nu = 0.000063
obj = -47.412747, rho = -0.706443
nSV = 222, nBSV = 0
Total nSV = 222
Accuracy = 98.5217% (5798/5885) (classification)
8 98.5217 (best c=0.25, rate=98.9295)
.........*......*
optimization finished, #iter = 15220
nu = 0.000031
obj = -47.412747, rho = -0.706443
nSV = 222, nBSV = 0
Total nSV = 222
Accuracy = 98.5217% (5798/5885) (classification)
9 98.5217 (best c=0.25, rate=98.9295)
.........*......*
optimization finished, #iter = 15220
nu = 0.000016
obj = -47.412747, rho = -0.706443
nSV = 222, nBSV = 0
Total nSV = 222
Accuracy = 98.5217% (5798/5885) (classification)
10 98.5217 (best c=0.25, rate=98.9295)
.....*...*
optimization finished, #iter = 8074
nu = 0.020558
obj = -45.260469, rho = -0.719830
nSV = 348, nBSV = 146
Total nSV = 348
Accuracy = 99.0683% (1914/1932) (classification)
Parameter C= 0.25 Accuracy = 99.0683
Classwise Accuracy for 6 = 99.2693 Classwise Accuracy for 8 = 98.8706

Q1(b)



Q1(c) **Parameter used and class-wise accuracy on the test set**:
Parameter C= 0.25 Accuracy = 99.0683
Classwise Accuracy for 6 = 99.2693 Classwise Accuracy for 8 = 98.8706

Q1(d) Plot of distances of correctly classified and incorrectly classified points



Distances of incorrectly classified points are close to 0 (in the range 0-0.75), whereas those of correctly classified points is much higher (mostly greater than 0.5)
This is expected as, more the distance of a point from the hyperplane, more is the confidence in classification and hence higher accuracy (more it belongs to the certain class). If the point is closer to hyperplane, it is more prone to misclassification

Q1(e) **RBF Kernel, grid search on all parameters to obtain a trained model:**

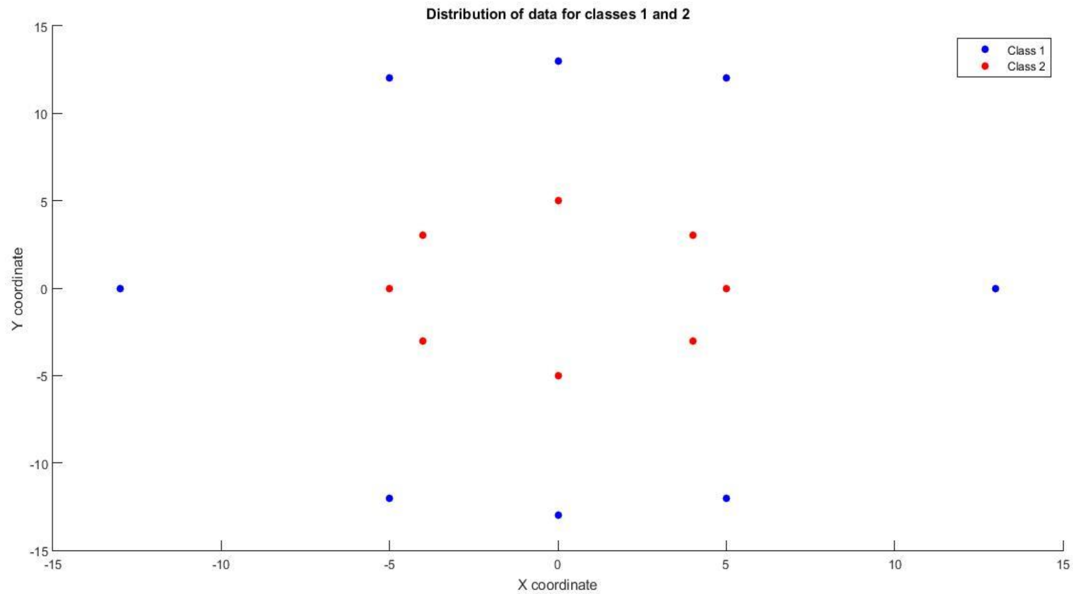
```
.......*....*
optimization finished, #iter = 11839
nu = 0.249982
obj = -2941.803159, rho = -0.005745
nSV = 5884, nBSV = 0
Total nSV = 5884
Accuracy = 50.2804% (2959/5885) (classification)
2 0 50.2804 (best c=4, g=0.0625, rate=99.6092)
.......*....*
optimization finished, #iter = 11839
nu = 0.249989
obj = -2941.901035, rho = -0.005778
nSV = 5884, nBSV = 0
Total nSV = 5884
Accuracy = 50.2804% (2959/5885) (classification)
2 1 50.2804 (best c=4, g=0.0625, rate=99.6092)
```

Final Parameters used and Classwise accuracy

```
....*.*
optimization finished, #iter = 5642
nu = 0.010567
obj = -497.451705, rho = 0.266230
nSV = 3661, nBSV = 0
Total nSV = 3661
Accuracy = 99.6377% (1925/1932) (classification)
Parameter C= 4 Parameter G= 0.0625 Accuracy = 99.6377
Classwise Accuracy for 6 = 99.2693 Classwise Accuracy for 8 = 100
```

Q1(f)

(a) Plot of points of the 2 classes to show distribution of classes



(b) **Yes**, it is possible to achieve 100% accuracy on the given dataset with the Kernel – RBF.
Parameters are: best c=0.5, g=0.0625, rate=100
Support Vectors - 100% accuracy achieved with Kernel RBF with above parameters.
Support vectors are <x,y>:

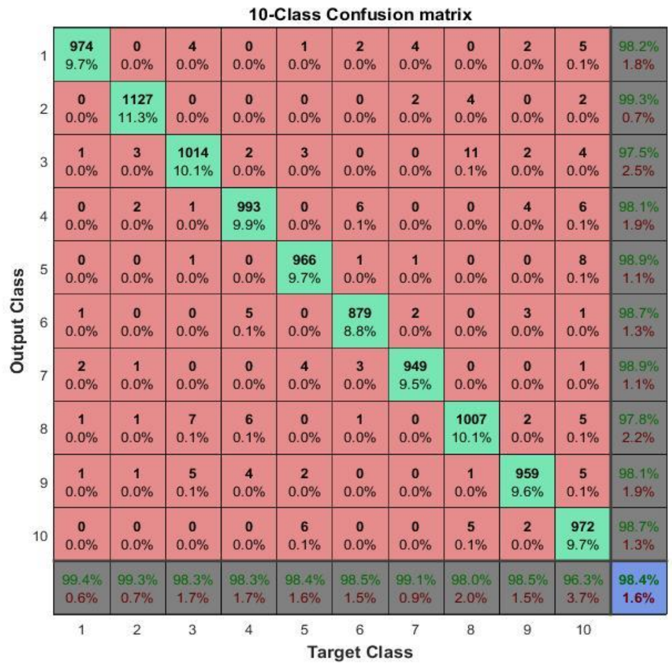
1. 0 13
2. 0 -13
3. 13 0
4. -13 0
5. 0 5
6. 0 -5
7. -5 0
8. 5 0

Q2(a) Implemented One-Versus-All multiclass SVM for the entire MNIST dataset (10 classes)

Classification Accuracy is : 98.4%

Classwise Accuracy is :
Class 0 : 99.3878%
Class 1 : 99.2952%
Class 2 : 98.2558%
Class 3 : 98.3168%
Class 4 : 98.3707%
Class 5 : 98.5426%
Class 6 : 99.0605%
Class 7 : 97.9572%

Class 8 : 98.46%
Class 9 : 96.333%
10-Class Confusion Matrix –



Classification is most accurate for classes 0,1,6 and least accurate for 7,9. Thus learnt classifiers predict accurately for 0,1,6 but not that accurately (or distance value from hyperplane is smaller) for 7,9

Q3(a) Using first 200 SPAM, HAM messages for training and rest for testing. Extracted meaningful features – **Most significant words were:** “call”, ”claim”, ”free”, ”get”, ”just”, ”now”, ”reply”, ”text”, ”txt”

Training using Sigmoid Kernel, applying grid search to find best parameters:

optimization finished, #iter = 518
nu = 0.253695
obj = -928127.073457, rho = -0.999836
nSV = 108, nBSV = 94
Total nSV = 108
Accuracy = 89.5805% (4634/5173) (classification)
10 0 91.591 (best c=2048, g=0.111111 rate=91.591)

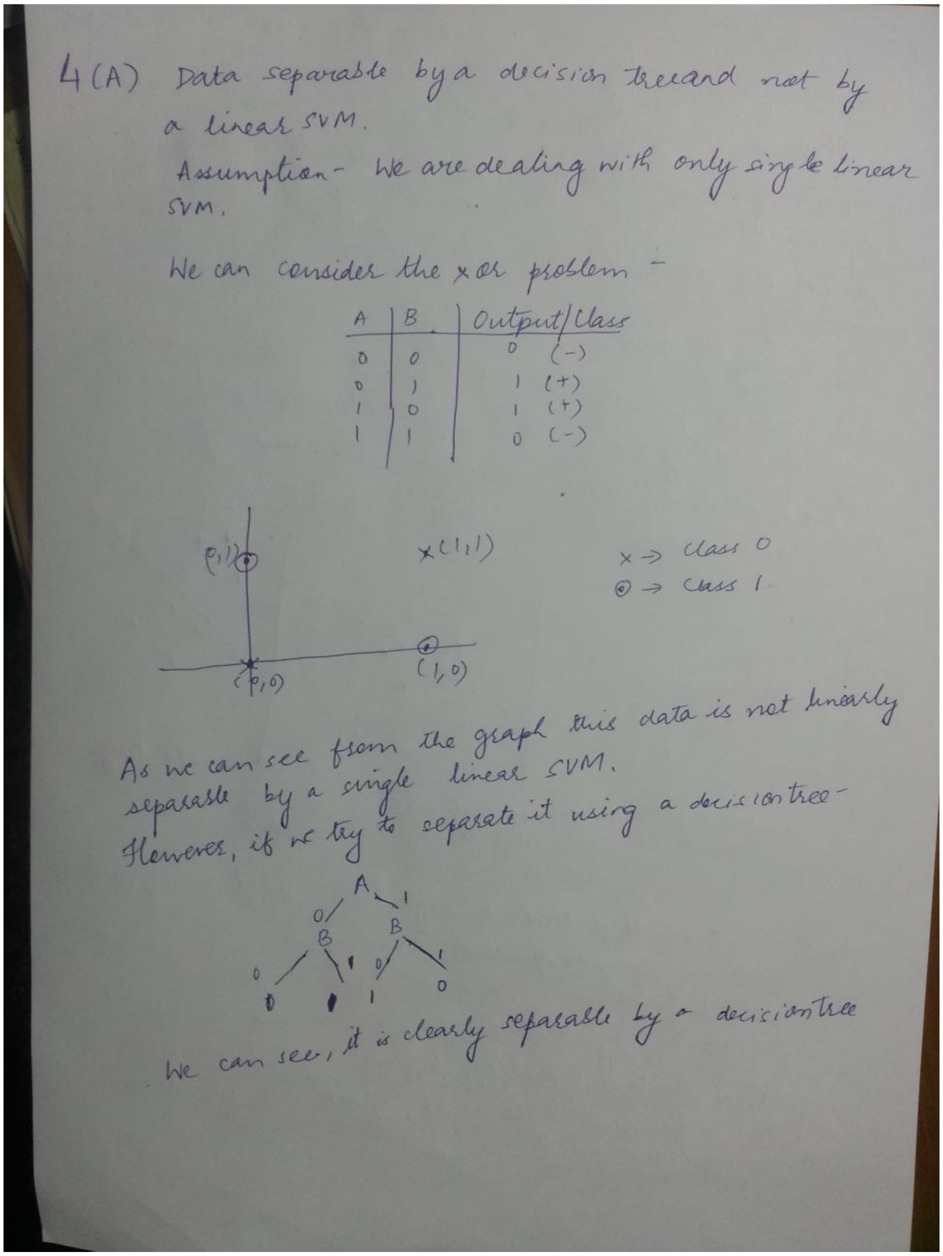
*
optimization finished, #iter = 1352
nu = 0.110942
obj = -98715290.561578, rho = -1.000034
nSV = 584, nBSV = 567
Total nSV = 584
Accuracy = 91.591% (4738/5173) (classification)

Parameter c= 2048 Parameter g= 0.11111 Accuracy = 91.591
Classwise Accuracy for ham = 96.4981 Classwise Accuracy for spam = 50.0914

Parameters chosen after grid search and classiwse accuracy on the test set:

Accuracy = 91.591% (4738/5173) (classification)
Parameter c= 2048 Parameter g= 0.11111 Accuracy = 91.591
Classwise Accuracy for ham = 96.4981 Classwise Accuracy for spam = 50.0914

Q4(a)



4(A) Data separable by a decision tree and not by a linear SVM.
Assumption - We are dealing with only single linear SVM.

We can consider the xor problem -

A	B	Output/Class
0	0	0 (-)
0	1	1 (+)
1	0	1 (+)
1	1	0 (-)

As we can see from the graph this data is not linearly separable by a single linear SVM.
However, if we try to separate it using a decision tree -

We can see, it is clearly separable by a decision tree

Q4(b)

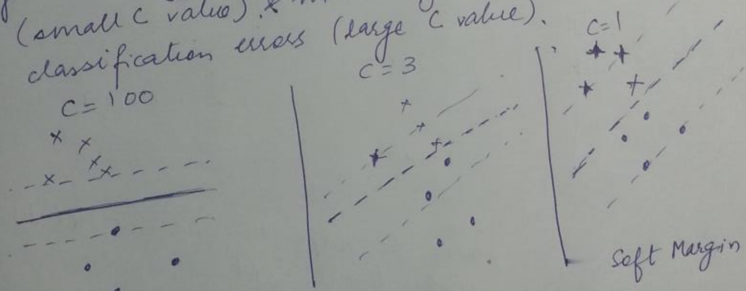
4 (B) For a 2 class problem, boundary of linear SVM obtained $c=0$ & $c=$ infinity will <u>not</u> be same.

There is no such example

Finding maximal margin corresponds to solving an optimization which involve minimizing the term $\frac{1}{2}\|w\|^2$ under the constraint that all exemplars are classified correctly.

Minimizing the term, $\frac{1}{2}\|w\|^2 + C\sum_i \xi_i$

The cost or penalty constant $C>0$ sets relative importance of maximizing margin & thus generalization performance (small C value) & minimizing amount of classification errors (large C value).



Hard Margin Better Generalization as we move left to right ⇒ Increase in error

Thus, $C=\infty$ would produce a single line - hard margin ξ ~~is going to produce~~ leading to 0 misclassifications. Whereas $C=0$ is going to produce a very wide margin ⇒ high misclassification. Since C is controlling tradeoff between error & margin width, depending on its value, it can never produce same margin width.

Q4(c)

4(c) $K_a(x,y)$, $K_b(x,y)$ > 2 kernels

$$\boxed{K_c(x,y) = K_a(x,y) * K_b(x,y)}$$

$K_c(x,y)$ is a kernel. Hence proposed statement is false.

Mathematical proof:

let ϕ_a be a feature map for K_a, ϕ_b be a feature map for K_b.

$$K_a(x,y) = \cancel{\phi_a(x).\phi_a(y)}\ \phi_a(x).\phi_a(y)$$

$$= \sum_{i=1}^{\infty} f_i(x)\, f_i(y)$$ {$f_i(x) \rightarrow$ i^{th} feature value under feature map ϕ_a}

$$K_b(x,y) = \phi_b(x)\, \phi_b(y)$$

$$= \sum_{j=1}^{\infty} g_j(y)\, g_j(x)$$ {$g_j(x) \rightarrow$ j^{th} feature value under feature map ϕ_b}

$$K_a(x,y)\, K_b(x,y) = (\phi_a(x)\, \phi_a(y))\, (\phi_b(x)\, \phi_b(y))$$

$$= \sum_{i=1}^{\infty} f_i(x)\, f_i(y) \sum_{j=1}^{\infty} g_j(x)\, g_j(y)$$

$$= \sum_{i,j} (f_i(x)\, g_j(x))\, f_i(y)\, g_j(y)$$

$\therefore$ We can define a feature map ϕ_c with a feature $h_{ij}(x) = f_i(x)\, g_j(x)$.

$$\Rightarrow K_a(x,y).\ K_b(x,y) = \phi_c(x)\, \phi(y) = K_c(x,y)$$

Hence Proved.