1. LIBSVM:

Results for different kernel and other default parameters. kernels vary with the parameter t

-t 0, -t 1, -t 2, -t 3

-t 0:

optimization finished, #iter = 579

nu = 0.017662

obj = -0.627017, rho = 1.172955

nSV = 40, nBSV = 0

Total nSV = 40

Accuracy = 85.7143% (30/35) (classification)

-t 1:

optimization finished, #iter = 162

nu = 0.022567

obj = -0.801149, rho = 0.404372

nSV = 57, nBSV = 0

Total nSV = 57

Accuracy = 74.2857% (26/35) (classification)

- t 2:

optimization finished, #iter = 99

nu = 0.801753

obj = -30.091940, rho = -0.076980

nSV = 71, nBSV = 22

Total nSV = 71

Accuracy = 77.1429% (27/35) (classification)

- t 3:

optimization finished, #iter:

nu: 0.957746

obj = -65.367107, rho = -0.492870

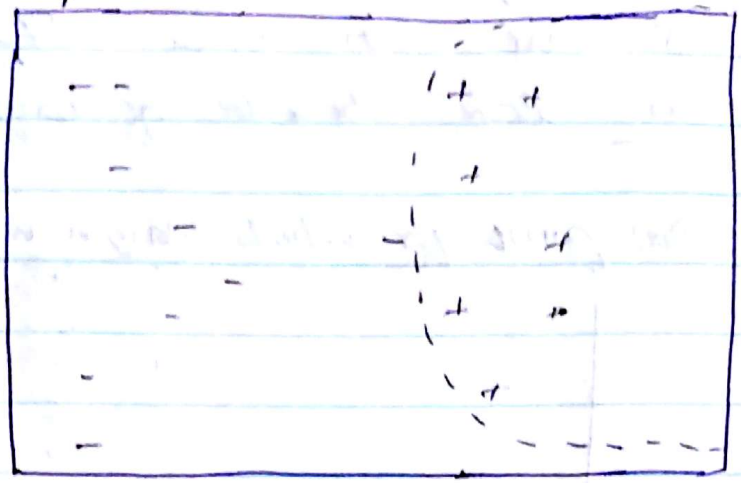
nSV: 68, nBSV = 68

Total nSV = 68

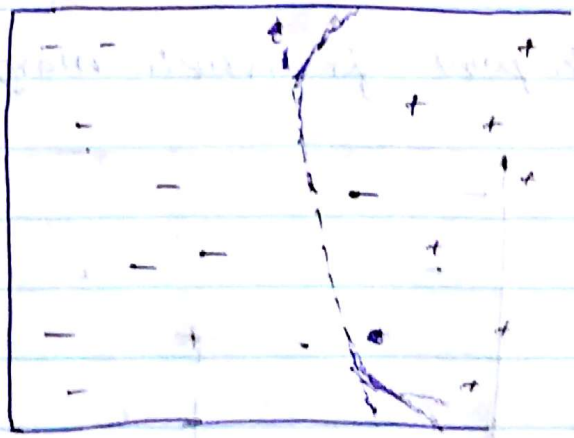
Accuracy: 45.7143% (16/35) (classification)

For Quadratic kernel

2. 1) For $C \to \infty$, penalty for misclassifying points is very high, so the decision boundary will perfectly separate the data.

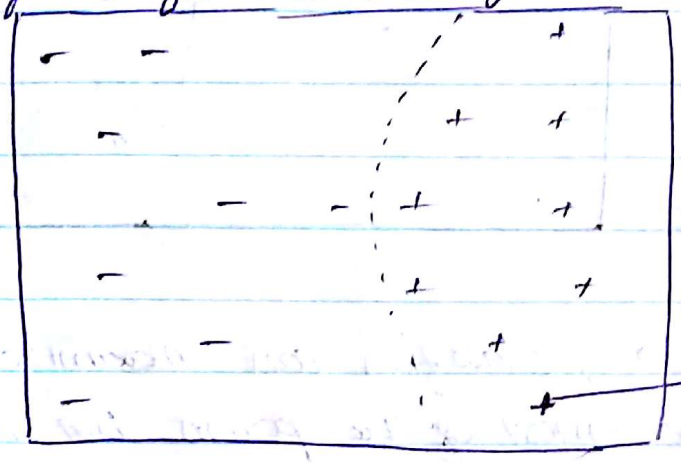


2) For $C = 0$, classifier can maximize the margin between most of the points but misclassifies few points because of low penalty

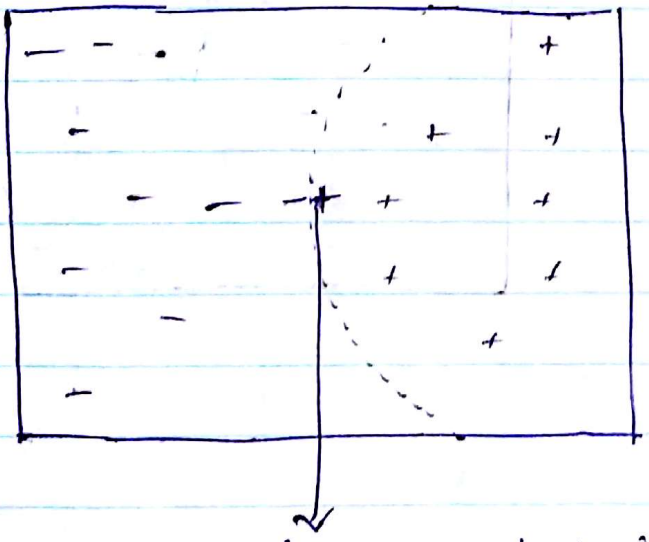


3. I think in first method where C value is high will classify data perfectly because of penalty value being high but it might be overfit model. So we can chose SECOND METHOD where C is low because of larger margins

4. Data point for which margin will not change

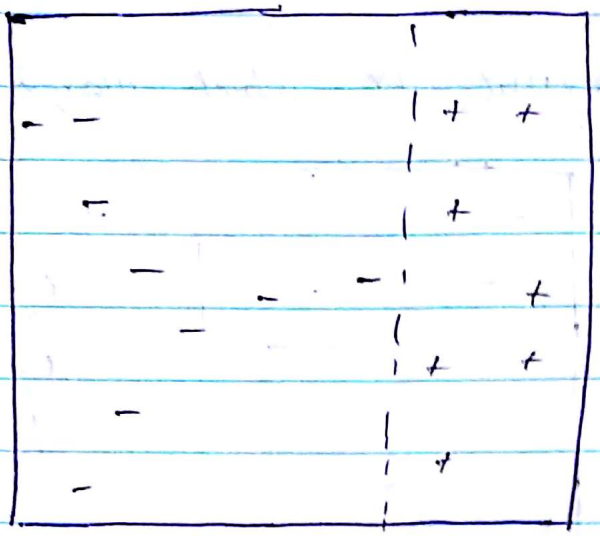


5. Data point for which margin will change

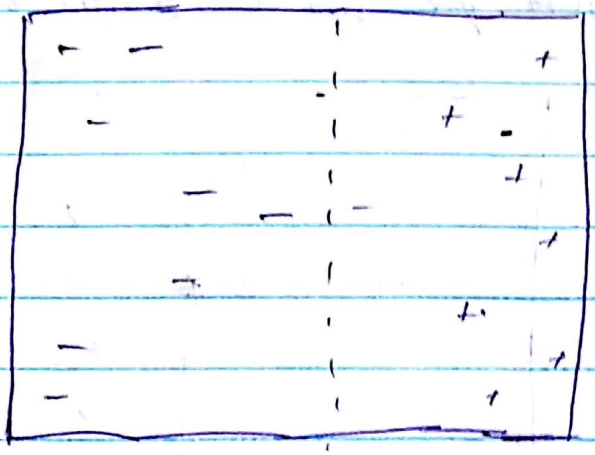


For Linear kernel

1) When $C \to \infty$, penalty for misclassification of points is high, so the decision boundary will perfectly separate the data.

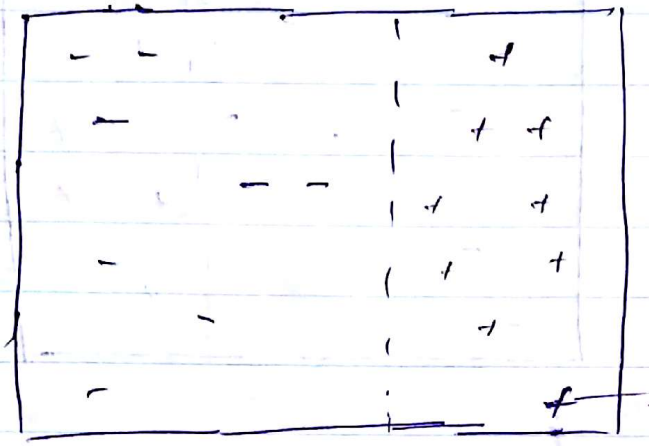


2) For $C = 0$, classifier can maximize the margin between most of the points but misclassifies few because of low priority



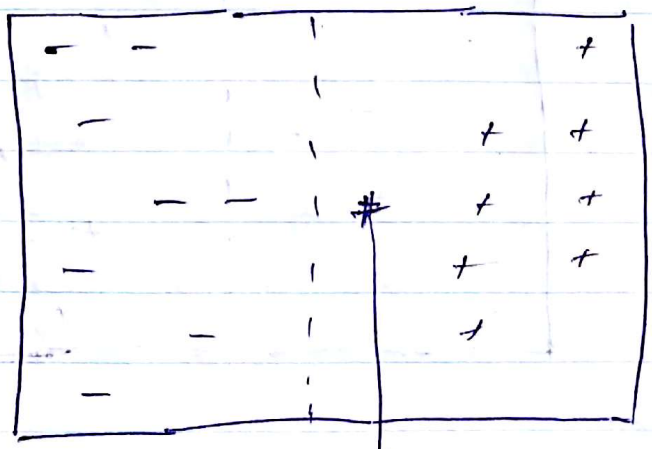
3. If C is high, it might overfit the classification, so it is better to choose C value as lower because of larger margins

4. Datapoint for which margin will not change



New data point for which margin will not change

5. Data point for which margin will change



New data point for which the margin change for larger values of C

3. 1) Euclidean distance:

$$d_E(x^i, x^j) = \sqrt{\sum_{k=1}^{p} (x_k^i - x_k^j)^2}$$

$x_1 : x_2 = x_3 = 0$ (Test data)

Red : $\sqrt{(0-0)^2 + (0-3)^2 + (0-0)^2} = \sqrt{9}$

Red : $\sqrt{(0-1)^2 + (0-0)^2 + (0-0)^2} = \sqrt{1}$

Red : $\sqrt{(0-0)^2 + (0-1)^2 + (0-3)^2} = \sqrt{10}$

Green : $\sqrt{(0-0)^2 + (0-1)^2 + (0-2)^2} = \sqrt{5}$

Green : $\sqrt{(0+1)^2 + (0-0)^2 + (0-1)^2} = \sqrt{2}$

Red : $\sqrt{(0-1)^2 + (0-0)^2 + (0-1)^2} = \sqrt{2}$

2) Classification when $k=1$

Test data is close to Red $\sqrt{1}$, So can be classified as red

3) Classification when $k=3$

Closest distant ones are $\sqrt{9}, \sqrt{5}, \sqrt{2}$ (Red, Green, Red). So prediction is red.

4) Training error when $k=1$ is 0

For any training example, its nearest neighbor in training set is always going to be itself. Split training set to trainset and validation set, Evaluate different values of k on validation set choose the one that has least error