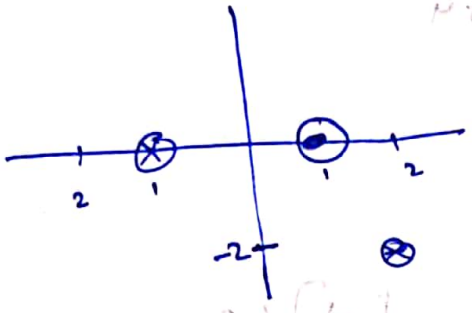
William Scott
MT18026.

AML Assignment 2

1)

(i) Given data.



$$X = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 2 & -2 \end{bmatrix}$$

a) $x_1 + x_2 = 0$

applying X on the given decision boundary

$$y = \begin{bmatrix} -1+0 \\ 1+0 \\ 2-2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} <0 \\ >0 \\ =0 \end{matrix}$$

as the decision boundary missed the point $(2,-2)$, the margin is not maximized.

b) $x_1 + 1.5x_2 = 0$

applying X on the given decision boundary

$$y = \begin{bmatrix} -1+0 \\ 1+0 \\ 2-3 \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \begin{matrix} <0 \\ >0 \\ <0 \end{matrix}$$

$$\text{margin width} = \frac{2}{\sqrt{1+2.25}} = 1$$

distance from each x.

$\frac{|v|}{\sqrt{\sum v_i^2}}$ → distence.

(-1,0) = 0.554

(1,0) = 0.554

(2,-2) = 0.554

c) $x_1 + 2x_2 = 0$

$$y = \begin{bmatrix} -1+0 \\ 1+0 \\ 2-2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ -2 \end{bmatrix} \begin{matrix} <0 \\ >0 \\ <0 \end{matrix}$$

margin width $= \frac{2}{\sqrt{4+4}} = 0.89$

distence from each x

$(-1,0) = \frac{1}{\sqrt{5}}$

$(1,0) = \frac{1}{\sqrt{5}}$

$(2,-2) = \frac{2}{\sqrt{5}}$

not maximized as (2,-2) is not a support vector.

d) $2x_1 + 3x_2 = 0$

$$y = \begin{bmatrix} -2+0 \\ 2+0 \\ 4-6 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ -2 \end{bmatrix} \begin{matrix} <0 \\ >0 \\ <0 \end{matrix}$$

margin $= \frac{2}{\sqrt{4+9}} = \frac{2}{\sqrt{13}} < 1$

not max margin.

(ii)

a) D

as D has oval shape, polynomial kernel degree 2 is capable of seperating the data points in oval form

b) B

as the lines are more curvy and are having multiple curves, only degree 3 polynomial can cause this decision boundary.

c) A

Sigma has inverse effect of gamma.

if sigma is less, more overfitting tends to happen.

as we consee from A, the model is more overfit. (than B)

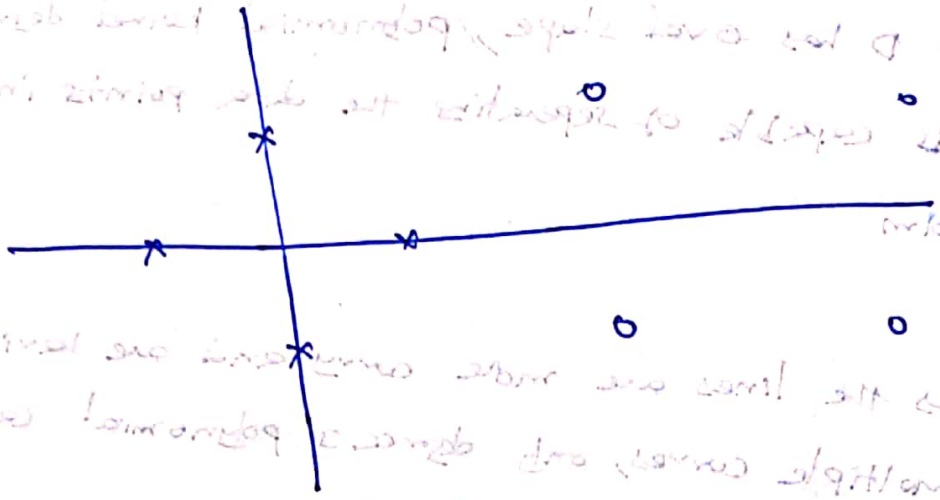
d) C

compared to A, C has less overfit. and with higher gamma less overfit happens.

Note:

rbf is not to be confused with linear as linear kernel seperates both the classes and not RBF, it can form seperate dimensions for each class.

4)



$$X = \begin{bmatrix} 3 & 1 \\ 6 & 1 \\ 3 & -1 \\ 6 & -1 \end{bmatrix} \quad Y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}$$

from the graph we can say that (3,1) (3,-1) and (1,0) are support vectors (using linear kernel)

taking average of the [0] class points (3,1) & (3,-1) we can consider a new support support vector

then support vectors are (3,0) & (1,0)

$$\vec{w}_2 = (3,0) - (1,0) = (2,0)$$

W.K.T

$y(\bar{w}x + b) = 1$

Substitute support vectors

$-1(2x + b) = 1$

$+1(3x + b) = 1$

$2x + b + 1 = 0$ ——— ①

$6x + b - 1 = 0$

Subtract above two

$-4x = -2$

$x = \frac{1}{2}$

$\Rightarrow w = [2(\frac{1}{2}), 0]$

$= [1, 0]$

Substitute w in ①

$2(\frac{1}{2}) + b + 1 = 0$

$1 + b + 1 = 0$

$b = -2$

$\Rightarrow$ **$w = [1, 0]$ $b = -2$**

5) Limitations of SVM on large dataset.

- for large scale datasets, the computational complexity is very high as the kernel computation and the decision boundary has to be computed for each and every datapoint.

with large datasets, it's highly likely that the kernel is non-linear and non-linear kernels add up more to complexity

- when SVM is computing the decision boundary, in the initial stages it needs all of the data points to be available in the memory. (ram)

and with large datasets, SVM takes up lot of memory usage too. [storing kernel matrix]

-adding more number of classes just increases the number of hyperplanes.

Computation Complexity:

$$O(\max(n,d)\min(n,d)^2)$$

n - no. of datapoints

d - dimensions of data points.

reference: "Training a support vector machine in the primal".

as we can see, the computation complexity increases a lot with datapoints.

- the computation increases quadratically.

for multiclass

1 vs all	one vs one
$O(L^4 C^3)$	$O(L^2 C^3)$

L - classes

C - datapoints

reference: mani's lecture pdf.

Solutions for SVM large dataset.

there have been multiple researches for improving SVM runtime.

abstract

- reduce features (not ideal)
- parallel SVM
- running multiclass classifiers for SVM in different cores.
- reformulate kernelised SVM as linear SVM
- Nystrom approximation: using eigen values.
- randomizing features.
- approximating optimizing problem with a set of smaller subproblems.

Solution Psuedocode

~~ref: Efficient support vector learning for large datasets.~~

ref: A Randomized algorithm for large Scale support vector learning.

randSVM-1 (D, k, r)

D - dataset

k - estimate of the number of support vectors

r - sample size = ck, c>0

① S = randomsubset (D, r);

// pick a random subset, S, of size r from dataset D

② sv = svmlearn (ϕ, S);

// sv - set of support vectors obtained by solving the problem S.

③ v = {x ∈ D - S | violates (x, sv)}

// violator - non sampled point not satisfying KKT conditions.

④ while |v|>0 and |sv|<k do:

⑤ R = randomsubset(v, r-|sv|);

// pick a random subset from the set of violators

⑥ sv = svmlearn(sv, R);

// sv - set of support vectors obtained from sv∪R.

⑦ v = {x∈D - (sv ∪ R) | violates(x, sv)};

⑧ end while

⑨ return sv.