1.1) Ridge Regression equation given as:-

$$\underset{w,b}{\text{minimize}} \; \lambda \|w\|^2 + \sum_{i=1}^{n} (w^T x_i + b - y_i)^2$$

The above equation in the matrix form is:-

$$\min \underbrace{(Y - w^T x)^T (Y - w^T x)}_{\text{Sum of squared error}} + \underbrace{\lambda w^T w}_{\text{Regularization term.}} \quad \text{— (1)}$$

To solve equation -(1) we need to take gradient w.r.t w and set that to zero(0).

$$-2X^T(Y - w^T X) + 2\lambda w = 0 \quad \left[\text{as } AB = B^T A^T \text{ and } \frac{\partial}{\partial A}(A^T A) = 2A\right]$$

$$\Rightarrow -2X^T Y + 2X^T X w + 2\lambda w = 0 \quad [\bar{X} = [X; 1_n^T]]$$

$$\Rightarrow w(2X^T X + 2\lambda I) = 2X^T Y$$

$$\Rightarrow w = c^{-1} d \quad [\text{as } c = XX^T + \lambda I, \; d = XY]$$

Hence,

$$\boxed{\bar{w} = \bar{c}^{-1} \bar{d}} \quad [\text{where, } \bar{w} = [w; b], \; \bar{c} = \bar{X}\bar{X}^T + \lambda \bar{I}, \; \bar{d} = \bar{X}Y]$$

(proved)

Scanned by CamScanner

1)

1-a)

Given, $C = XX^T + \lambda I_{k+1}$

$$\Rightarrow C = \lambda I_{k+1} + [\bar{x}_1\ \bar{x}_2 \dots \bar{x}_n][\bar{x}_1\ \bar{x}_2 \dots \bar{x}_n]^T$$

$$= \lambda I_{k+1} + [\bar{x}_1\ \bar{x}_2 \dots \bar{x}_n]\begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_n^T \end{bmatrix}$$

$$\Rightarrow C = \lambda I_{k+1} + \sum_{j=1}^{n} \bar{x}_j \bar{x}_j^T \quad \text{—— (1)}$$

Now, if we remove ith element, eq-(1) gives us:-

$$C_{(i)} = \lambda I_{k+1} + \sum_{j=1, j\neq i}^{n} \bar{x}_j \bar{x}_j^T$$

$$\Rightarrow \boxed{C_{(i)} = C - \bar{x}_i \bar{x}_i^T}$$

Ans

Now, $d = \bar{x} y$

$$= [x_1\ x_2 \dots x_n][y_1\ y_2 \dots y_n]^T$$

$$= [x_1\ x_2 \dots x_n]\begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix}$$

$$\Rightarrow d = \sum_{j=1}^{n} x_j y_j \quad \text{—— (2)}$$

If we remove ith element from eq-(2) then,

$$d_{(i)} = \sum_{j=1; j\neq i}^{n} x_j y_j$$

$$\Rightarrow \boxed{d_{(i)} = d - x_i y_i}$$

Ans:

Scanned by CamScanner

1)

1-3) From previous answers we get that

$$C_{(i)} = (C - x_i x_i^T)$$

$$\Rightarrow C_{(i)}^{-1} = (C - x_i x_i^T)^{-1} \quad \text{— (1)}$$

By using Sherman-Morrison formula we can rewrite equation (1) as:

$$\boxed{C_{(i)}^{-1} = C^{-1} + \frac{C^{-1} x_i x_i^T C^{-1}}{1 - x_i^T C^{-1} x_i}}$$

Answer

[By using Sherman-Morrison formula]

Scanned by CamScanner

1)

1.2)

Now, given that

$$w = c^{-1}d \quad \text{and} \quad w_{(i)} = c_{(i)}^{-1} d_{(i)}$$

$$= (c - x_i x_i^T)^{-1}(d - x_i y_i)$$

[from 9.1.2 soln]

Now, using Sherman-Morrison formula:

$$w_{(i)} = \left(c^{-1} + \frac{c^{-1}x_i x_i^T c^{-1}}{1 - x_i^T c^{-1} x_i}\right)(d - x_i y_i)$$

$$= c^{-1}d - c^{-1}x_i y_i + \frac{c^{-1}x_i x_i^T c^{-1} d - c^{-1}x_i x_i^T c^{-1} x_i y_i}{1 - x_i^T c^{-1} x_i}$$

$$= w + c^{-1}x_i\left(-y_i + \frac{x_i^T c^{-1} d - x_i^T c^{-1} x_i y_i}{1 - x_i^T c^{-1} x_i}\right)$$

$$= w + c^{-1}x_i\left(\frac{-y_i + \cancel{y_i x_i^T c^{-1} x_i} + x_i^T c^{-1} d - \cancel{x_i^T c^{-1} x_i y_i}}{1 - x_i^T c^{-1} x_i}\right)$$

$$= w + c^{-1}x_i \frac{(-y_i + x_i^T w)}{1 - x_i^T c^{-1} x_i}$$

$$\Rightarrow \bar{w}_{(i)} = \bar{w} + c^{-1}\bar{x}_i\left[\frac{-y_i + \bar{x}_i^T \bar{w}}{1 - \bar{x}_i^T c^{-1} \bar{x}_i}\right]$$

Proved

Scanned by CamScanner

1)

1.5)

$$w_{(i)}^T x_i - y_i = x_i^T w_{(i)} - y_i$$ [by using linear algebra of matrix $A^TB = B^TA$]

$$= x_i^T \left[w + C^{-1} x_i \frac{(-y_i + x_i^T w)}{1 - x_i^T C^{-1} x_i} \right]$$

$$= \frac{x_i^T w (1 - x_i^T C^{-1} x_i) + x_i^T C^{-1} x_i (-y_i + x_i^T w) - y_i + y_i x_i^T C^{-1} x_i}{(1 - x_i^T C^{-1} x_i)}$$

$$= \frac{w^T x_i - x_i^T w x_i^T C^{-1} x_i - x_i^T C^{-1} x_i y_i + x_i^T C^{-1} x_i x_i^T w - y_i + y_i x_i^T C^{-1} x_i}{(1 - x_i^T C^{-1} x_i)}$$

$$= \frac{w^T x_i - y_i}{1 - x_i^T C^{-1} x_i} - \frac{x_i^T C^{-1} x_i (\cancel{w \cdot x_i^T} + \cancel{y_i} - x_i^T \cancel{w} - \cancel{y_i})}{1 - x_i^T C^{-1} x_i}$$

[using $A^TB = B^TA$]

$$\Rightarrow w_{(i)}^T x_i - y_i = \frac{w^T x_i - y_i}{1 - x_i^T C^{-1} x_i}$$

$$\Rightarrow \bar{w}_{(i)}^T \bar{x}_i - y_i = \frac{\bar{w}^T x_i - y_i}{1 - \bar{x}_i^T C^{-1} \bar{x}_i}$$ (proved)

Scanned by CamScanner

1)

1.6) We already know that,

$$\bar{w} = c^{-1}d \;\Big|\; c^{-1} = \bar{X}X^T + \lambda I \;\Big|\; d = \bar{X}y$$

breaking this into smaller operations :-

$$XX^T = \text{multi}(d\times n, n\times d) = d^2n$$

$$XX^T + \lambda I = \text{add}(d\times d, d\times d) = d^2$$

$$c^{-1} = d^3$$

$$\bar{X}y = \text{multi}(d\times n, n\times 1) = nd$$

$\Rightarrow$ Total complexity $= \underline{O(nd^2 + d^3)}$

$\Rightarrow$ For n-samples $= \underline{O(n^2d^2 + nd^3)}$ (Answer)

Again for the equation in question 1.5

$$\bar{w}^T\bar{x}_i - y_i = \frac{\bar{w}^T x_i - y_i}{1 - \bar{x}_i^T c^{-1}\bar{x}_i}$$

breaking this into smaller parts :-

$$\bar{w}^T\bar{x}_i = \text{multi}(1\times d, d\times n) = d$$

$$\bar{x}_i^T c^{-1} = \text{multi}(1\times d, d\times d) = d^2$$

$$\bar{x}_i^T c^{-1}\bar{x}_i = \text{multi}(d\times d, d\times 1) = d$$

So for c^{-1} and $\bar{w}$ the order will be $\underline{O(d^2)}$

(Answer)

Scanned by CamScanner

2.1) Given, function to be used for Naive Bayes

$X \to Y$ where,

$x = (x_1, x_2)$ [x_1 is boolean variable and x_2 is a continuous variable]

$Y =$ boolean variable.

Now, let Y is distributed by binomial distribution with,

$$P(Y=1) = \nu$$

$$P(Y=0) = 1 - P(Y=1) = (1-\nu)$$

Now, by using bayes theorem,

$$P(Y=1|x) = \frac{P(Y=1).P(x|Y=1)}{P(Y=1).P(x|Y=1) + P(Y=0).P(x|Y=0)}$$

$$= \frac{1}{1 + \exp\left(\ln \frac{P(Y=0).P(x|Y=0)}{P(Y=1).P(x|Y=1)}\right)}$$

$$\Rightarrow P(Y=1|x) = \frac{1}{1 + \exp\left(\ln \frac{1-\nu}{\nu} + \sum_i \ln \frac{P(x_i|Y=0)}{P(x_i|Y=1)}\right)} \quad \text{--- (1)}$$

Scanned by CamScanner

Now, <u>parameters of x_1</u>

i) $\theta_{11} = P(x_1 = 1 | Y=1)$ —— (2)

so, $P(x_1 = 0 | Y=1) = 1 - \theta_{11}$ —— (3)

(ii) $\theta_{10} = P(x_1 = 1 | Y=0)$ —— (4)

$P(x_1 = 0 | Y=0) = 1 - \theta_{10}$ —— (5)

Now, x_1 follows binomial distribution so,

$$\ln \frac{P(x_1 | Y=0)}{P(x_1 | Y=1)} = \ln \left(\frac{\theta_{10}^{x_1}}{\theta_{11}^{x_1}} \cdot \frac{(1-\theta_{10})^{(1-x_1)}}{(1-\theta_{11})^{(1-x_1)}} \right)$$

—— (6)

<u>Parameters for x_2</u>

Let, $P(x_2 | Y = y_k)$ is a Gaussian distribution of the form $N(\mu_{2k}, \sigma_{2k})$ [μ_{2k} is mean, σ_{2k} is std.]

where, σ_{2k} depends on the value of k for y.

$$\ln \left(\frac{P(x_2 | Y=0)}{P(x_2 | Y=1)} \right) = \ln \frac{\frac{1}{\sqrt{2\pi}\,\sigma_{20}} \exp\left(-\frac{(x_2 - \mu_{20})^2}{2\sigma_{20}^2}\right)}{\frac{1}{\sqrt{2\pi}\,\sigma_{21}} \exp\left(-\frac{(x_2 - \mu_{21})^2}{2\sigma_{21}^2}\right)}$$

Scanned by CamScanner

$$\Rightarrow \ln \frac{P(x_2|Y=0)}{P(x_2|Y=1)} = \ln \frac{\sigma_{21}}{\sigma_{20}} + \frac{(x_2-\mu_{21})^2}{2\sigma_{21}^2} - \frac{(x_2-\mu_{20})^2}{2\sigma_{20}^2} \quad \text{— (7)}$$

From- equation — (1)

$$P(Y=1|x) = \frac{1}{1+\exp\left(\ln\frac{(1-\nu)}{\nu} + \sum_i \ln \frac{P(x_i|Y=0)}{P(x_i|Y=1)}\right)}$$

$$= \frac{1}{1+\exp\left(\ln\frac{(1-\nu)}{\nu} + \ln\frac{P(x_1|Y=0)}{P(x_1|Y=1)} + \ln\frac{P(x_2|Y=0)}{P(x_2|Y=0)}\right)}$$

$$= \frac{1}{1+\exp\left(\ln\frac{(1-\nu)}{\nu} + \ln\frac{\theta_{10}^{x_1}(1-\theta_{10})^{(1-x_1)}}{\theta_{11}^{x_1}(1-\theta_{11})^{(1-x_1)}} + \ln\frac{\sigma_{21}}{\sigma_{20}} + \frac{(x_2-\mu_{21})^2}{2\sigma_{21}^2} + \frac{(x_2-\mu_{20})^2}{2\sigma_{20}^2}\right)}$$

$$= \frac{1}{1+\exp\left(\ln\frac{(1-\nu)}{\nu} + \frac{x_1\cdot\theta_{10}(1-\theta_{11})}{\theta_{11}(1-\theta_{10})} + \ln\frac{(1-\theta_0)}{(1-\theta_{11})} + \ln\frac{\sigma_{21}}{\sigma_{20}} + \frac{\mu_{21}^2\sigma_{20}^2-\mu_{20}^2\sigma_{21}^2}{2\sigma_{20}^2\sigma_{21}^2} + \frac{\mu_{20}\sigma_{21}^2-\mu_{21}\sigma_{20}^2}{\sigma_{20}^2\sigma_{21}^2}x_2 + \right.}$$

Scanned by CamScanner

$$\frac{\sigma_{20}^2 - \sigma_{21}^2}{2\sigma_{20}^2\sigma_{21}^2} x_2^2 \Big)$$

By substituting,

$$w_0 = \ln\frac{(1-\pi)}{\pi} + \ln\left(\frac{1-\theta_0}{1-\theta_{11}}\right) + \frac{\mu_{21}^2\sigma_{20}^2 - \mu_{20}^2\sigma_{21}^2}{2\sigma_{20}^2\sigma_{21}^2} + \ln\frac{\sigma_{21}}{\sigma_{20}}$$

$$w_1 = \ln\frac{\theta_{10}(1-\theta_{11})}{\theta_{11}(1-\theta_{10})}$$

$$w_2 = \ln\frac{\mu_{20}\sigma_{21}^2 - \mu_{21}\sigma_{20}^2}{\sigma_{20}^2\sigma_{21}^2}, \quad v_1 = \frac{\sigma_{20}^2 - \sigma_{21}^2}{2\sigma_{20}^2\sigma_{21}^2}$$

Since, $\sigma_{21} \neq \sigma_{20}$, Hence quadratic term will not disappear. The form of $P(Y=1|x)$ is no longer a logistic regression form.

so,

$$P(Y=1|x) = \frac{1}{1+\exp(w_0 + w_1x_1 + w_2x_2 + v_1x_2^2)}$$

and

$$P(Y=0|x) = 1 - P(Y=1|x)$$

$$\Rightarrow P(Y=0|x) = \frac{\exp(w_0 + w_1x_1 + w_2x_2 + v_1x_2^2)}{1+\exp(w_0 + w_1x_1 + w_2x_2 + v_1x_2^2)}$$

Scanned by CamScanner

2.2>

For,

$$P(Y=1|x) = \frac{1}{1+\exp\left(-\sum_{i=1}^{d} \theta_i x_i + \theta_{d+1}\right)}$$

Since, Y is boolean and $X = (X_1, \ldots, X_d)$ is a vector of boolean variables and it is a gaussian Naive bayes classifier where each x_i is a binomial distribution.

By applying Bayes theorem.

$$P(Y=1|x) = \frac{P(Y=1) \cdot P(X|Y=1)}{P(Y=1) \cdot P(X|Y=1) + P(Y=0) \cdot P(X|Y=0)}$$

$$= \frac{1}{1+\exp\left(\ln \frac{P(Y=0) \cdot P(X|Y=0)}{P(Y=1) \cdot P(X|Y=1)}\right)}$$

$$= \frac{1}{1+\exp\left(\ln \frac{(1-\nu)}{\nu} + \sum_i \ln \frac{P(x_i|Y=0)}{P(x_i|Y=1)}\right)} \quad \text{—(1)}$$

where, $\nu = P(Y=1)$

and $(1-\nu) = P(Y=0)$

Scanned by CamScanner

As x_i ~~and~~ follows binomial distribution, we get

$$P(x_i|Y=0) = \theta_{i0}^{x_i}(1-\theta_{i0})^{(1-x_i)} \quad \text{——— (2)}$$

$$P(x_i|Y=1) = \theta_{i1}^{x_i}(1-\theta_{i1})^{(1-x_i)} \quad \text{——— (3)}$$

Replacing (2) & (3) to equation — (1)

$$P(Y=1|x) = \frac{1}{1+\exp\left(\ln\frac{1-\nu}{\nu} + \sum_{i=1}^{d}\frac{\theta_{i0}^{x_i}(1-\theta_{i0})^{(1-x_i)}}{\theta_{i1}^{x_i}(1-\theta_{i1})^{(1-x_i)}}\right)}$$

$$= \frac{1}{1+\exp\left(\ln\frac{1-\nu}{\nu} + \sum_{i=1}^{d} x_i \ln\frac{\theta_{i0}}{\theta_{i1}} + (1-x_i)\ln\frac{(1-\theta_{i0})}{(1-\theta_{i1})}\right)}$$

$$= \frac{1}{1+\exp\left(\underbrace{\ln\frac{1-\nu}{\nu} + \frac{(1-\theta_{i0})}{(1-\theta_{i1})}}_{w_0} + \underbrace{\sum_{i=1}^{d} x_i\left[\ln\frac{\theta_{i0}}{\theta_{i1}} - \ln\frac{(1-\theta_{i0})}{(1-\theta_{i1})}\right]}_{w_i}\right)}$$

Scanned by CamScanner

Now, substituting,

$$w_0 = \ln \frac{1-K}{K} + \sum_{i=1}^{d} \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})} \text{ and}$$

$$w_i = \ln \frac{\theta_{i0}}{\theta_{i1}} - \ln \frac{(1-\theta_{i0})}{(1-\theta_{i1})}$$

we get,

$$P(Y=1|x) = \frac{1}{1+\exp\left(\sum_{i=1}^{d} w_i x_i + w_0\right)}$$

and

$$P(Y=0|x) = 1 - P(Y=1|x)$$

$$= \frac{\exp\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)}{1+\exp\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)}$$

The above is same as logistic regression formulation. We also know that logistic regression is the discriminative ~~convergent~~ counterpart of Naive Bayes, based on conditional independence assumption.

Scanned by CamScanner

(3)

3.)

(1) Given dual objective function for Kernel SVM can be written as a quadratic programming problem in the below formulation:-

$$\underset{Z}{\text{maximize}} \quad \frac{1}{2} Z^T H Z + f^T Z$$

such that,

$$A \cdot Z \leq b$$
$$A_{eq} Z = b_{eq}$$
$$lb \leq Z \leq ub$$

where, H, A and A_{eq} are matrices and f, b, b_{eq}, lb and ub are vectors. Also,

H = Semi positive definite matrix of dimension $p \times p$

A = $m \times p$ matrix, $[0, \ldots, 0]$

f = p-column vector, $[-1, \ldots, 1]^T$

b = m-column vector

$A_{eq} = [y_1, \ldots, y_n]$

$b_{eq} = 0$

$lb = [0, \ldots, 0]$

$ub = [\infty, \ldots, \infty]$

Scanned by CamScanner

3.1

2.
Implemented SVM dual objective using MATLAB quadprog function and the code for the same can be found here:

/code/Kernel_SVM_3_1/svm_main.m

3.

As we are using Linear Kernel to optimize the SVM here so the code for computing the W is given the below mentioned MATLAB script:

/code/Kernel_SVM_3_1/compute_kernel.m

Code for computing b is given in the below MATLAB script:

/code/Kernel_SVM_3_1/compute_b0.m

4.
Execution of Linear Kernel SVM with C = 0.1

C = 0.100000000000000
train_accuracy = 0.972375690607735
eval_accuracy = 0.953678474114441
obj_val_train = 352
obj_val_eval = 350
Number of Support Vectors = 339
Confusion Matrix = **/Result/confusion_matrix_01.csv**

5.

Execution of Linear Kernel SVM with C = 10

C = 10
train_accuracy = 1
eval_accuracy = 0.978201634877384
obj_val_train = 362
obj_val_eval = 359
Number of Support Vectors = 123
Confusion Matrix = **/Result/confusion_matrix_10.csv**

3.2>

1.>

Subgradient of L_i w.r.t to W_{y_i}

$$L_i = \frac{1}{2n} \sum_{j=1}^{k} \|W_j\|^2 + cL(W, x_i, y_i)$$

$$= \frac{1}{2n} \sum_{j=1}^{k} \|W_j\|^2 + c\left(\max\{W_{y_i}^T x_i - W_{y_i}^T x_i + 1, 0\}\right)$$

$$\frac{\partial L_i}{\partial W_{y_i}} = \begin{cases} \frac{1}{n}\|W_j\| + c(-x_i), & \text{when, } W_{y_i}^T x_i > W_{y_i}^T x_i + 1 \\ \frac{1}{n}\|W_j\| + 0, & \text{when, } W_{y_i}^T x_i + 1 > W_{y_i}^T x_i \end{cases}$$

[for $j = y_i$]

3.2>

2

For, $j = y_i$, $$\frac{\partial L_i}{\partial W_{y_i}} = \begin{cases} \frac{1}{n}\|W_j\| + c(x_i), & \text{when, } W_{y_i}^T x_i + 1 > W_{y_i}^T x_i \\ \frac{1}{n}\|W_j\| + 0, & \text{when } 1 + W_{y_i}^T x_i < W_{y_i}^T x_i \end{cases}$$

Scanned by CamScanner

3.2)

(3) If $j \neq y_i$ and $j \neq \hat{y}_i$ then,

$$w_{\hat{y}_i}^T x_i - w_{y_i}^T x_i + 1 = 0$$

then,

$$\frac{\partial L_i}{\partial w_j} = \frac{1}{n} \| w_j \| + 0$$

Scanned by CamScanner

3.2

4.
Implemented SGD for multiclass SVM given in Algorithm 1, with tunable parameters eta0 and eta1. Initially started with weight matrix W as 0.

The MATLAB script for this is implementation is **/code/MC_SVM_3_2/svm_main.m**

5.

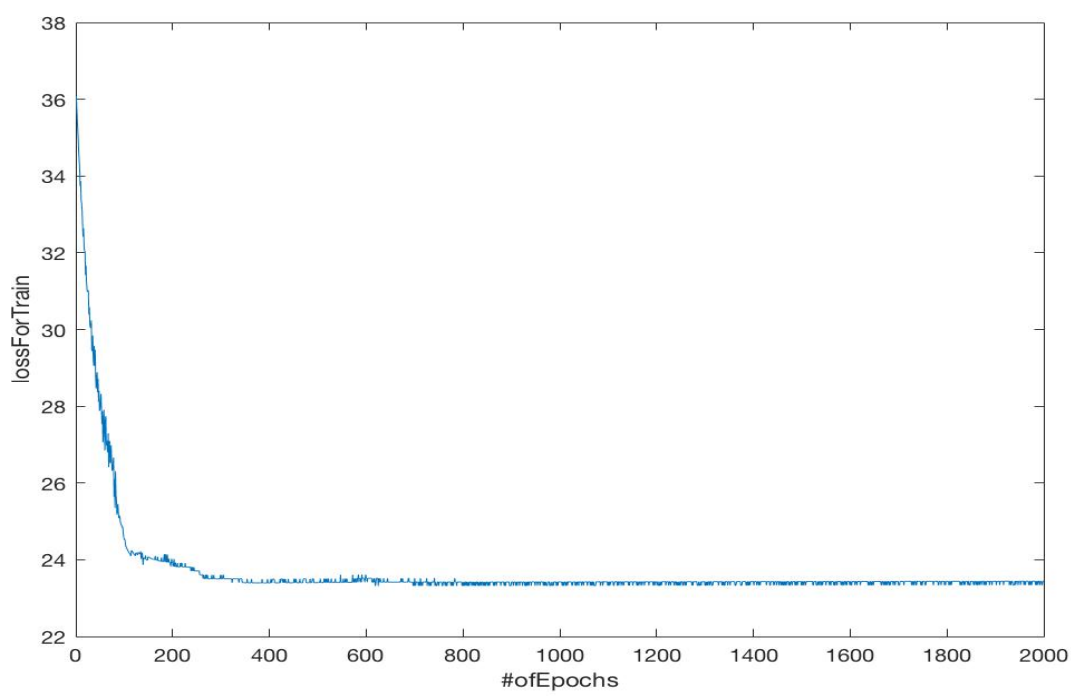
a) **Execution of SDG SVM on training data with eta0 = 1, eta1 = 100 and C = 0.1**

maxEpoch = 2000
eta0 = 1
eta1 = 100
C = 0.100000000000000

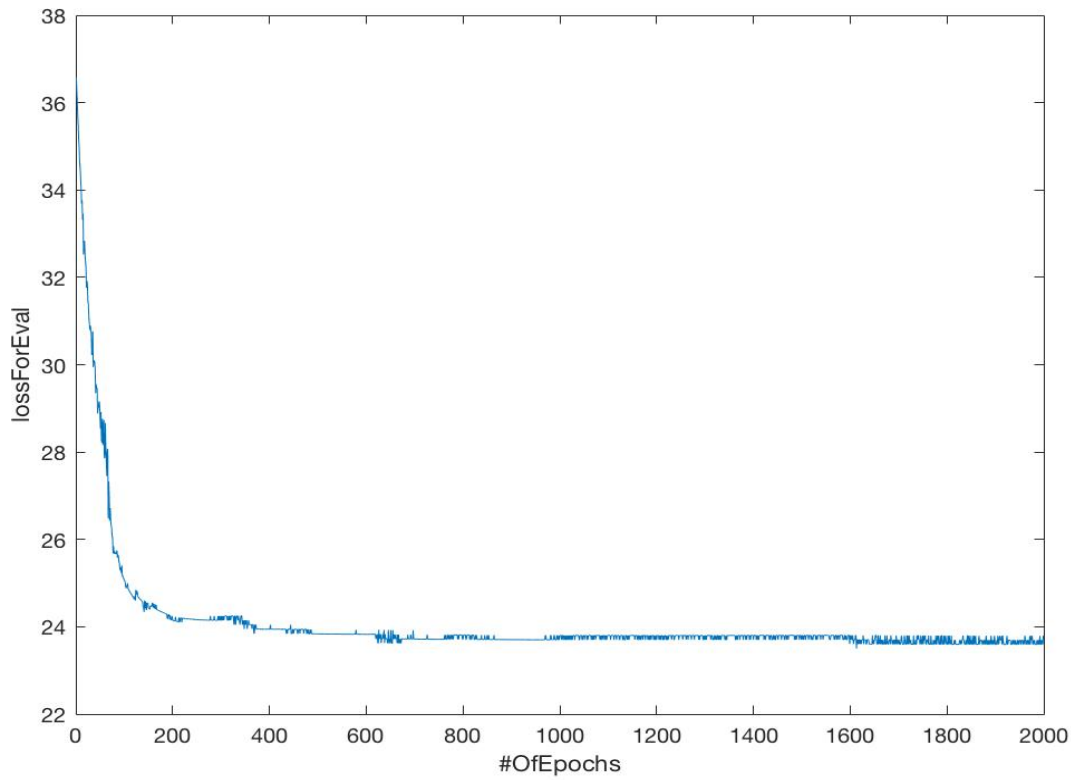
train_accuracy = 0.961325966850829
train_obj_val = 362

eval_accuracy = 0.956403269754768
eval_obj_val = 367

Plot of Loss vs Epoch on train dataset from q3_1_data.mat



Plot of Loss vs Epoch on validation dataset from q3_1_data.mat



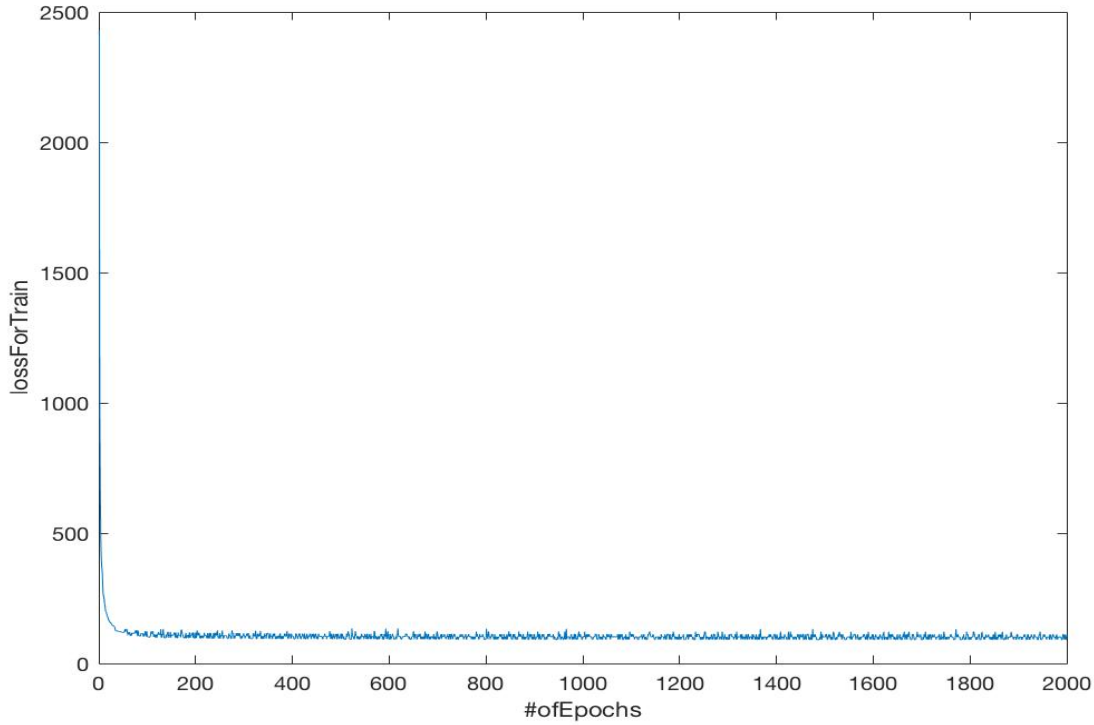
b) Execution of SDG SVM on training data with eta0 = 1, eta1 = 100 and C = 10

maxEpoch = 2000
eta0 = 1
eta1 = 100
C = 10

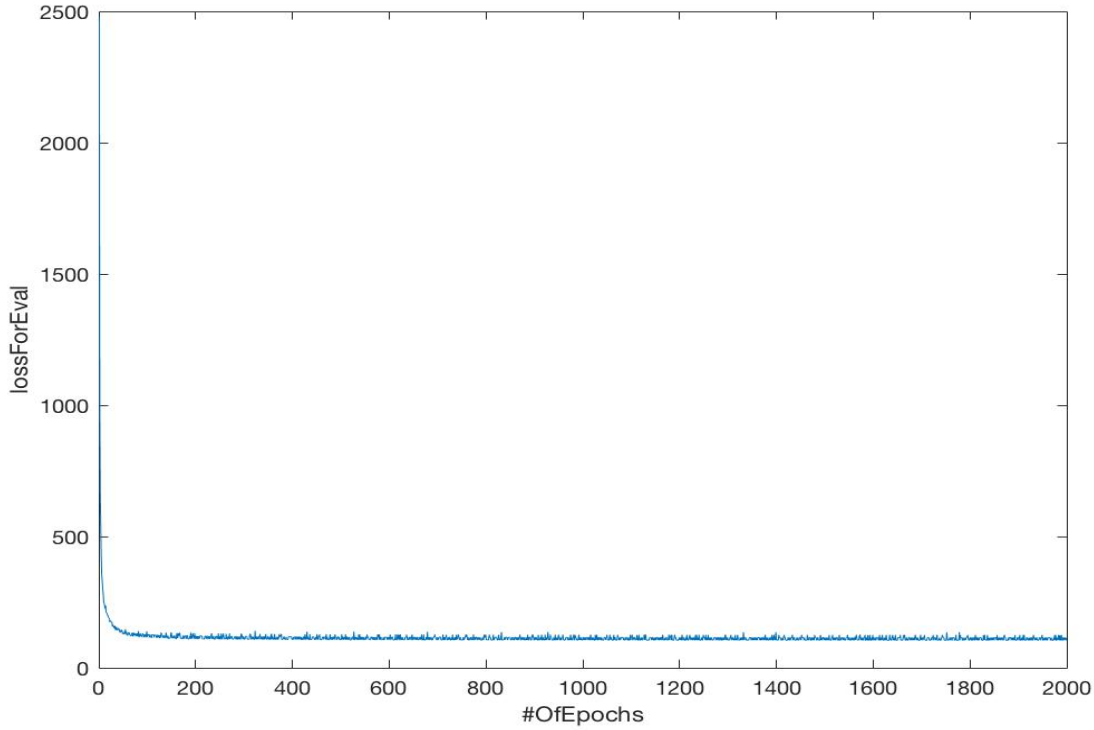
train_accuracy = 0.994475138121547
train_obj_val = 362

eval_accuracy = 0.991825613079019
eval_obj_val = 367

Plot of Loss vs Epoch on train dataset from q3_1_data.mat



Plot of Loss vs Epoch on validation dataset from q3_1_data.mat



c) **Objective value comparison:**

SVM	C = 0.1 on Training data	C = 10 on Training data	C = 0.1 on Validation data	C = 10 on Validation data
Linear Kernel	352	362	350	359
SGD Hinge Loss	362	362	367	367

6.

a) **The prediction error on valD, valLb in q3 1 data.mat**

<u>C = 0.1</u>
The prediction error for C = 0.1 can be found in file:
/Result/eval_loss_history_C_01.xlsx

<u>C = 10</u>
The prediction error for C = 10 can be found in file:
/Result/eval_loss_history_C_10.xlsx

b) **The prediction error on trD, trLb**

<u>C = 0.1</u>
The prediction error for C = 0.1 can be found in file:
/Result/train_loss_history_C_01.xlsx

<u>C = 10</u>
The prediction error for C = 10 can be found in file:
/Result/ train_loss_history_C_10.xlsx

c) **Sum of Wj ^ 2**

- On valD, valLb in q3 1 data.mat

<u>C = 0.1</u>
sumW_eval = 19.8552371272575

<u>C = 10</u>
sumW_eval = 149.576254560492

- On trD, trLb data

<u>C = 0.1</u>
sumW_train = 19.3018248021065

<u>C = 10</u>
sumW_train = 143.042004818712

7.

The MATLAB script for this is implementation is
/code/Kaggle_HL_3_2_7/svm_main.m

For the ‘q3 2 data.mat’ data, to achieve the best accuracy, I have used the below configuration and got the training loss accordingly:

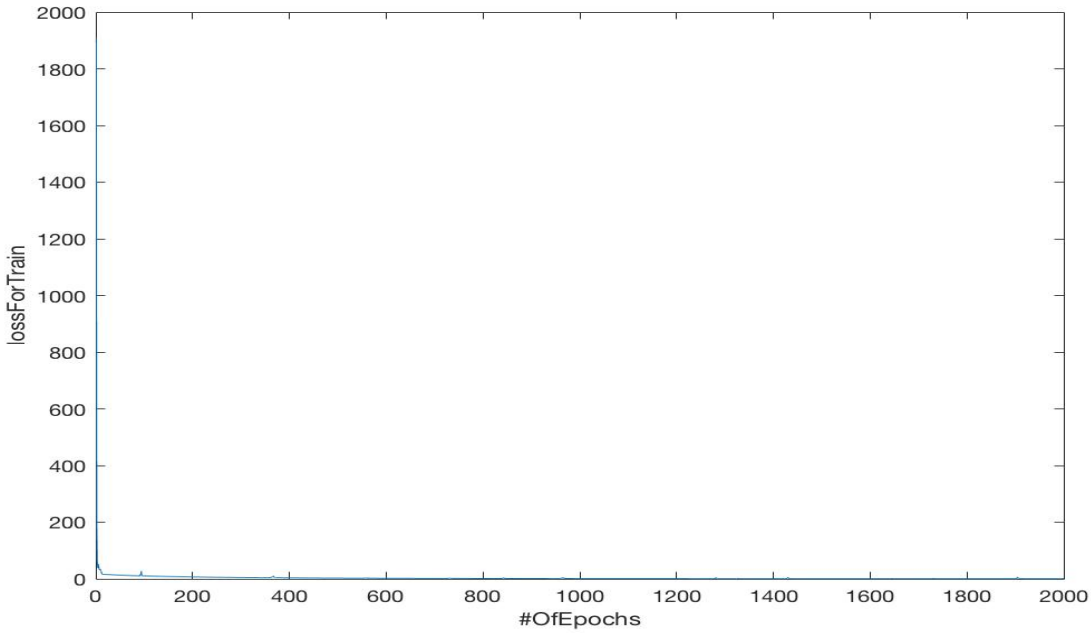
C = 0.500000000000000
eta0 = 1
eta1 = 300
maxEpoch = 2000

train_accuracy = 1
train_obj_val = 7930

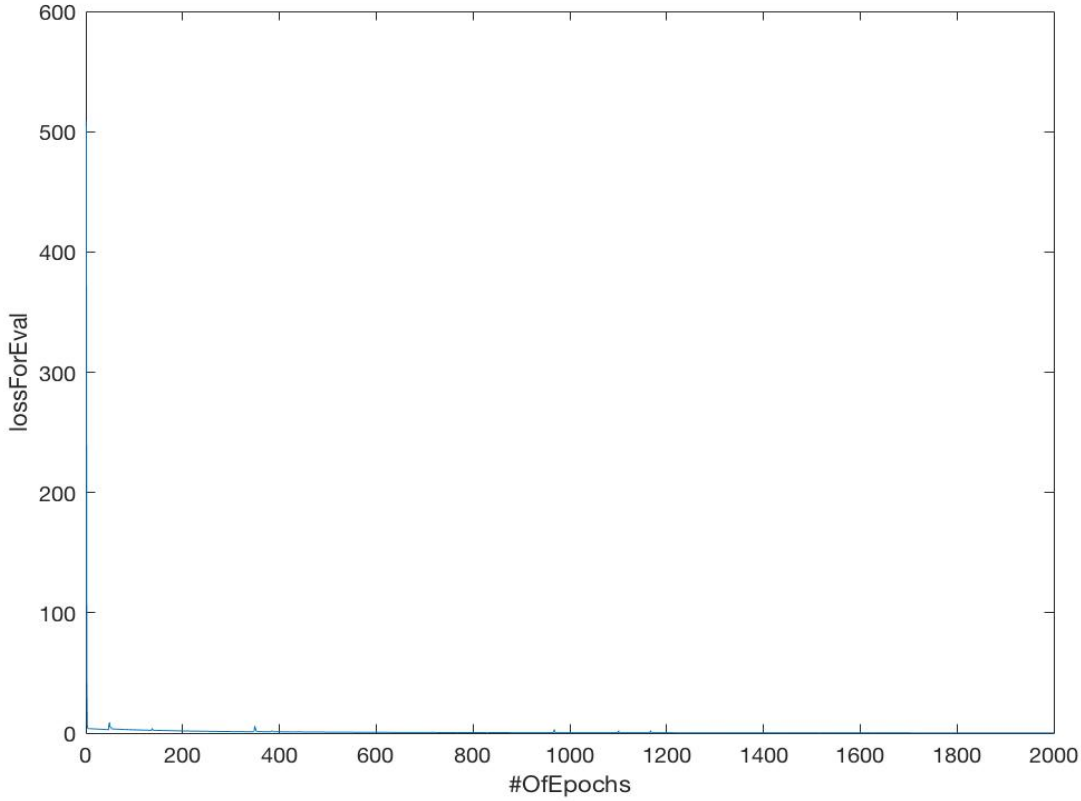
eval_accuracy = 1
eva-obj_val = 2120

sumW_train = 1.17816072864544
sumW_eval = 0.251408694479636

<u>Plot of Loss vs Epoch on train dataset</u>



Plot of Loss vs Epoch on validation dataset



Final Kaggle Submission file can be found here:

/Result/Submission_170048888.csv

I have used my Kaggle name ‘gourabbhattacharyya’ to participate in the competition.

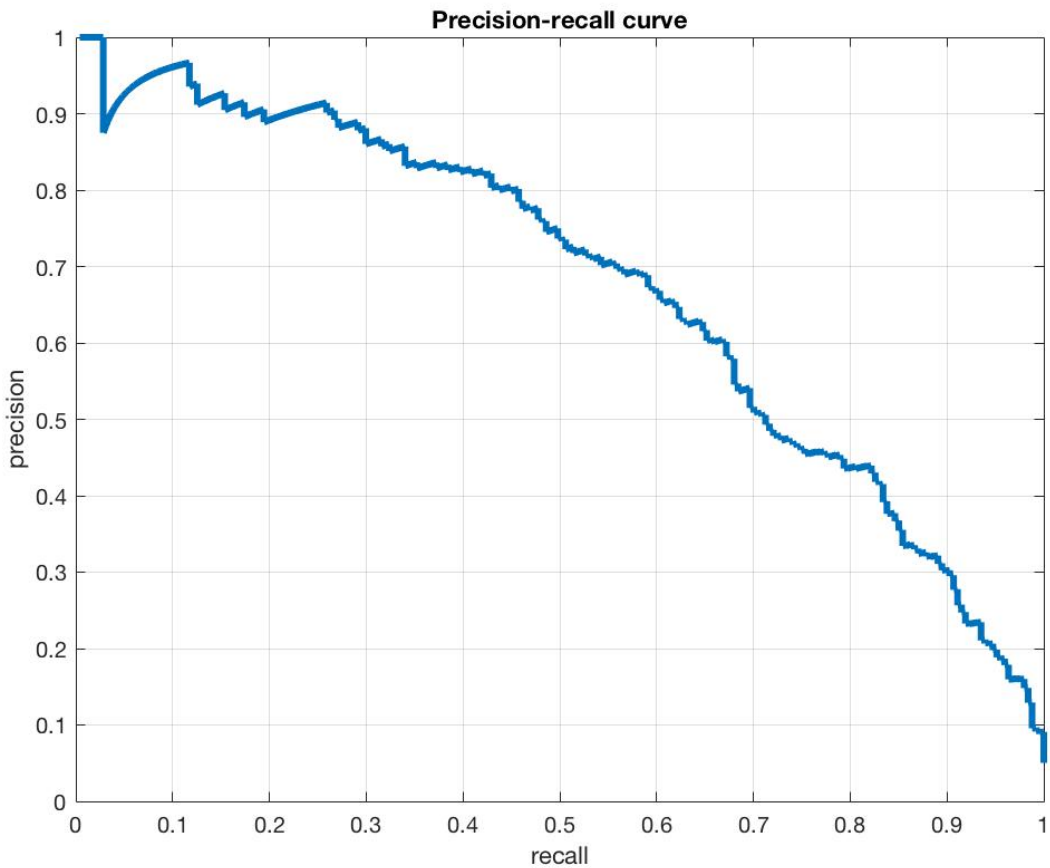
55	▾ 25	**gourabbhattacharyya**		0.60188	6

4.4

1.

I have generated the training data using HW2_Utils.getPosAndRandomNeg and then trained using the SVM (implemented in 3.2). Then I have used HW2_Utils.genRsltFile to generate the result file. Finally, used HW2_Utils.cmpAP on the result file to get AP and precision recall curve.

Script for this activity could be found at: **/code/prec-recall_plot_4_4_1/execute.m**

AP value for validation data = 0.679701646923219

precision recall curve on validation data