CSE 512 - Machine Learning HW 2

111462188 – Mihir Chakradeo

Q.1 Ridge Regression and LOOCV

QUESTION 1 - RIDGE REGRESSION AND LOOCV.

$$\min_{w,b} \quad \lambda \|w\|^2 + \sum_{i=1}^{n} (w^T x_i + b - y_i)^2.$$

(1.1) $\bar{w} = [w; b]$, $\bar{X} = [X; 1_n^T]$, $\bar{I} = [I_k, 0_k; 0_k^T, 0]$, $C = \bar{X}\bar{X}^T + \lambda \bar{I}$
$d = \bar{X}y$. Show that the solution of Ridge Regression is : $\bar{w} = C^{-1}d$.

Ans.

$$\min_{w,b} \quad \lambda \|\bar{w}\|^2 + \sum_{1}^{n} (\bar{w}^T \bar{x}_i \cancel{+b} - y_i)^2.$$

$$\Rightarrow \min_{w,b} \quad \lambda \|\bar{w}\|^2 + \|\bar{X}^T\bar{w} \cancel{+b} - Y\|_2^2$$

Take gradient wrt $\bar{w}$ and set to 0.

$$2\lambda\bar{w} + 2\bar{X}(\bar{X}^T\bar{w} - Y) = 0.$$

$$\bar{X}(\bar{X}^T\bar{w} - Y) + \lambda\bar{w} = 0$$

$$\bar{X}\bar{X}^T\bar{w} - \bar{X}Y + \lambda\bar{w} = 0.$$

$$(\bar{X}\bar{X}^T + \lambda I)\bar{w} = \bar{X}Y$$

$$\therefore \bar{w} = (\bar{X}\bar{X}^T + \lambda I)^{-1} \bar{X} Y \qquad \left\{\text{pre multiply by } (XX^T + \lambda I)^{-1}\right\}$$

$$\boxed{\bar{w} = C^{-1}d}$$

(1.2) Now suppose we remove x_i from the training data, let $C_i, d_i, \bar{w}_i$ be the corresponding matrices for removing x_i. Express C_i in terms of C and x_i. Express d_i in terms of d and x_i.

Ans. (i) We know : $C = \bar{X}\bar{X}^T + \lambda I.$

Removing $x_i \Rightarrow$ $C_i = \bar{X}\bar{X}^T - \bar{x}_i\bar{x}_i^T + \lambda I$

$$\boxed{C_i = C - \bar{x}_i\bar{x}_i^T}$$

(ii) we know : $d = \bar{X}y$

$\therefore d_i = (\bar{X} - \bar{x}_i)y_i = \bar{X}y - \bar{x}_i y_i = \boxed{d - \bar{x}_i y_i}$

(1.3) Express C_i^{-1} in terms of C^{-1} and x_i.

Ans. ~~Using the~~ $C_i = C - \bar{x}_i \bar{x}_i^T$

$\therefore C_i^{-1} = (C - \bar{x}_i \bar{x}_i^T)^{-1}$

Using the Sherman-Morisson formula:

$$(A+uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}.$$

$$\therefore \boxed{C_i^{-1} = C^{-1} + \frac{C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}}$$

(1.4) To show That: $\bar{w}_i = \bar{w} + (C^{-1}\bar{x}_i)\dfrac{-y_i + \bar{x}_i^T\bar{w}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}$

Ans. We have: $\bar{w} = C^{-1}d$

$$\therefore \bar{w}_i = C_i^{-1} d_i$$

$$= \left(C^{-1} - \frac{C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}\right)(\bar{X}\bar{Y} - \bar{x}_i y_i)$$

$$= C^{-1}(\bar{X}\bar{Y} - \bar{x}_i y_i) - \left(\frac{C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}\right)(\bar{X}\bar{Y} - \bar{x}_i y_i)$$

$$= C^{-1}d - C^{-1}\bar{x}_i y_i - \left(\frac{C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}\right)(\bar{X}\bar{Y} - \bar{x}_i y_i)$$

$$= -C^{-1}\bar{x}_i y_i + \bar{w} - \left(\frac{C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}}{1-\bar{x}_i^TC^{-1}\bar{x}_i}\right)(\bar{X}\bar{Y} - \bar{x}_i y_i)$$

$$= \bar{w} - \frac{C^{-1}\bar{x}_i y_i(1-\bar{x}_i^TC^{-1}\bar{x}_i) - C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}(\bar{X}\bar{Y} - \bar{x}_i y_i)}{1-\bar{x}_i^TC^{-1}\bar{x}_i}$$

$$= \bar{w} - \frac{C^{-1}\bar{x}_i y_i - C^{-1}\bar{x}_i y_i \bar{x}_i^TC^{-1}\bar{x}_i - C^{-1}\bar{x}_i\bar{x}_i^TC^{-1}(\bar{X}\bar{Y} - \bar{x}_i y_i)}{1-\bar{x}_i^TC^{-1}\bar{x}_i}$$

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

$$= \bar{w} + \left[\frac{-c^{-1}\bar{x_i}y_i + c^{-1}\bar{x_i}\,\bar{x_i}^T c^{-1}(\bar{x_i}y_i + \bar{x}y - \bar{x_i}y_i)}{1-\bar{x_i}^T c^{-1}\bar{x_i}}\right]$$

$$= \bar{w} + \left[\frac{-c^{-1}\bar{x_i}y_i + c^{-1}\bar{x_i}\,\bar{x_i}^T w}{1-\bar{x_i}^T c^{-1}\bar{x_i}}\right]$$

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

(1.5) Show that : LOOCV error for removing the i^{th} training data is:

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

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

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

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

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

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

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

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

(1.6) Formula from 1.5:

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

- Time ~~complexity~~ for calculating: $C^{-1} = k^3$.
- Time taken to calc. $\bar{w}^T x_i - y_i = k$
- Time taken to calc. $1 - \bar{x}_i^T C^{-1} \bar{x}_i = k^2$.

$\therefore$ Overall complexity $= \boxed{O(nk^2 + k^3)}$

$\rightarrow$ Usual way of computing LOOCV: $\boxed{O(nk^3)}$

$\therefore$ we can see that formula from (1.5) is faster.

Q.2 Naïve Bayes and Logistic Regression

(21) Give formula for computing $P(Y|X)$.

$Y \to$ boolean

$X = (X_1, X_2)$, $X_1 \to$ boolean ; $X_2 \to$ continuous

Ans. Let $P(X_1|Y=0) = \lambda_0^{X_1}(1-\lambda_0)^{1-X_1}$ {as X_1 is boolean}

$P(X_1|Y=1) = \lambda_1^{X_1}(1-\lambda_1)^{1-X_1}$.

Also, let $P(X_2|Y=0) = \frac{1}{\sqrt{2\pi}\cdot\sigma_0} e^{-\frac{(X_2-\mu_0)^2}{2\sigma_0^2}}$

$P(X_2|Y=1) = \frac{1}{\sqrt{2\pi}\,\sigma_1} e^{-\frac{(X_2-\mu_1)^2}{2\sigma_1^2}}$

we know that: $P(Y|X) = \frac{P(X|Y)P(Y)}{\sum_y P(X|Y=y)P(Y=y)}$.

$$\therefore P(Y=1|X) = \frac{\lambda_1^{X_1}(1-\lambda_1)^{1-X_1} \cdot \frac{1}{\sqrt{2\pi}\,\sigma_1} e^{-\frac{(X_2-\mu_1)^2}{2\sigma_1^2}} \cdot P(Y=1)}{\lambda_1^{X_1}(1-\lambda_1)^{1-X_1}\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(X_2-\mu_1)^2}{2\sigma_1^2}} \cdot P(Y=1) + \lambda_0^{X_1}(1-\lambda_0)^{1-X_1}\frac{1}{\sqrt{2\pi}\,\sigma_0} e^{-\frac{(X_2-\mu_0)^2}{2\sigma_0^2}}}.$$

~~We can write for any (Y=y) as follows:~~

$$P(Y=0|X) = \frac{\lambda_0^{X_1}(1-\lambda_0)^{1-X_1}\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(X_2-\mu_0)^2}{2\sigma_0^2}} P(Y=0)}{\lambda_0^{X_1}(1-\lambda_0)^{1-X_1}\frac{1}{\sqrt{2\pi}\,\sigma_0} e^{-\frac{(X_2-\mu_0)^2}{2\sigma_0^2}} P(Y=0) + \lambda_1^{X_1}(1-\lambda_1)^{1-X_1}\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(X_2-\mu_1)^2}{2\sigma_1^2}}}.$$

(2.2)

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

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

$$= \frac{1}{1 + \left[\exp\left(\log\left(\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}\right)\right)\right]}$$

$$= \frac{1}{1 + \left(\exp\left(\log\left(\frac{1-\gamma}{\gamma}\right) + \sum_{1}^{d} \log\left(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)\right)\right)}$$

Consider following term:

$$\sum_{1}^{d} \log\left(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right) = \sum_{1}^{d}\left[\log P(X_i|Y=0) - \log P(X_i|Y=1)\right]$$

Let $\lambda_{i0} = P(X_i|Y=0)$, $\lambda_{i1} = P(X_i|Y=1)$

we can write $P(X_i|Y=0) = \lambda_{i0}^{X_i}(1-\lambda_{i0})^{1-X_i}$

and $P(X_i|Y=1) = \lambda_{i1}^{X_i}(1-\lambda_{i1})^{1-X_i}$

$$\therefore \sum_{1}^{d}\left[\log P(X_i|Y=0) - \log P(X_i|Y=1)\right] = \sum_{1}^{d}\left[\log\left(\lambda_{i0}^{X_i}(1-\lambda_{i0})^{1-X_i}\right) - \log\left(\lambda_{i1}^{X_i}(1-\lambda_{i1})^{1-X_i}\right)\right]$$

$$= \sum_{1}^{d}\left\{\left(X_i\log\lambda_{i0} + (1-X_i)\log(1-\lambda_{i0})\right) - \left(X_i\log\lambda_{i1} + (1-X_i)\log(1-\lambda_{i1})\right)\right\}$$

$$= \sum_{1}^{d}\left[X_i\log\lambda_{i0} + \log(1-\lambda_{i0}) - X_i\log(1-\lambda_{i0}) - X_i\log\lambda_{i1} - \log(1-\lambda_{i1}) + X_i\log(1-\lambda_{i1})\right]$$

$$= \sum_{1}^{d}\left\{X_i\left[\log\lambda_{i0} - \log(1-\lambda_{i0}) - \log(\lambda_{i1}) + \log(1-\lambda_{i1})\right] + \log(1-\lambda_{i0}) - \log(1-\lambda_{i1})\right\}$$

$$\therefore \sum_{1}^{d} \log\left(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right) = \sum_{1}^{d} \theta_i X_i + \theta_{d+1}$$

where : $\theta_i = \log \lambda_{i0} - \log(1-\lambda_{i0}) - \log \lambda_{i1} + \log(1-\lambda_{i1})$

$\theta_{d+1} = \sum_{1}^{d} \log(1-\lambda_{i0}) - \log(1-\lambda_{i1})$

$$\therefore \boxed{P(Y=1|X) = \frac{1}{1+\exp\left(-\left(\sum_{1}^{d}\theta_i X_i + \theta_{d+1}\right)\right)}}$$

Q.3 Implementation of SVM

QUESTION 3 - IMPLEMENTATION OF SVMs

(3.1)

(1) Write svm dual objective as a quadratic program.
Write down $H, f, A, b, Aeq, beq, lb, ub$.

$$\max_{\alpha} \sum_{j=1}^{n} \alpha_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \alpha_i y_j \alpha_j K(x_i, x_j)$$

$$\text{s.t.} \quad \sum_{j=1}^{n} y_j \alpha_j = 0.$$

$$0 \le \alpha_j \le C \quad \forall j.$$

Ans.

- $H = \text{diag}(Y) \cdot K(x_i, x_j) \cdot \text{diag}(Y)$

$$\underset{n \times n}{\begin{bmatrix} y_{11} & & 0 \\ & \ddots & \\ 0 & & y_{nn} \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} x_1 x_1 & \dots & x_1 x_n \\ \vdots & & \\ x_n x_1 & \dots & x_n x_n \end{bmatrix}} \underset{n \times n}{\begin{bmatrix} y_{11} & & 0 \\ & \ddots & \\ 0 & & y_{nn} \end{bmatrix}}$$

- $f = \underset{n \times 1}{\begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}}$

- $A = [\,]$
- $b = 0$

- $Aeq = Y^T \quad [y_1\ y_2 \dots y_n]_{1 \times n}$

- $beq = 0.$

- $lb = \underset{n \times 1}{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}$
- $ub = C \quad \underset{n \times 1}{\begin{bmatrix} C1 \\ \vdots \\ Cn \end{bmatrix}}$.

(3.1.4) and (3.1.5)

For C = 0.1

a. Confusion Matrix =

168	16
3	180

b. Accuracy = 0.9482
c. Objective value of the SVM = 24.7648
d. Support Vectors = 339

For C = 10

a. Confusion Matrix =

178	6
4	179

b. Accuracy = 0.9728
c. Objective value = 112.1461
d. Support Vectors = 126

(3.2) Implement Multiclass SVM using Stochastic Gradient Descent

(3.2) Multiclass SVM using SGD —

(1) Subgradient of L_i wrt w_{y_i} :-

$$\frac{\partial L_i}{\partial w_{y_i}} = \begin{cases} \frac{1}{n} w_{y_i} - C x_i, & \text{if } w_{\hat{y}_i}^T x_i - w_{y_i}^T x_i + 1 > 0. \\ \frac{1}{n} w_{y_i}, & \text{otherwise} \end{cases}$$

(2) Subgradient of L_i wrt $w_{\hat{y}_i}$:-

$$\frac{\partial L_i}{\partial w_{\hat{y}_i}} = \begin{cases} \frac{1}{n} w_{\hat{y}_i} + C x_i, & \text{if } w_{\hat{y}_i}^T x_i - w_{y_i}^T x_i + 1 > 0 \\ \frac{1}{n} w_{\hat{y}_i}, & \text{otherwise} \end{cases}$$

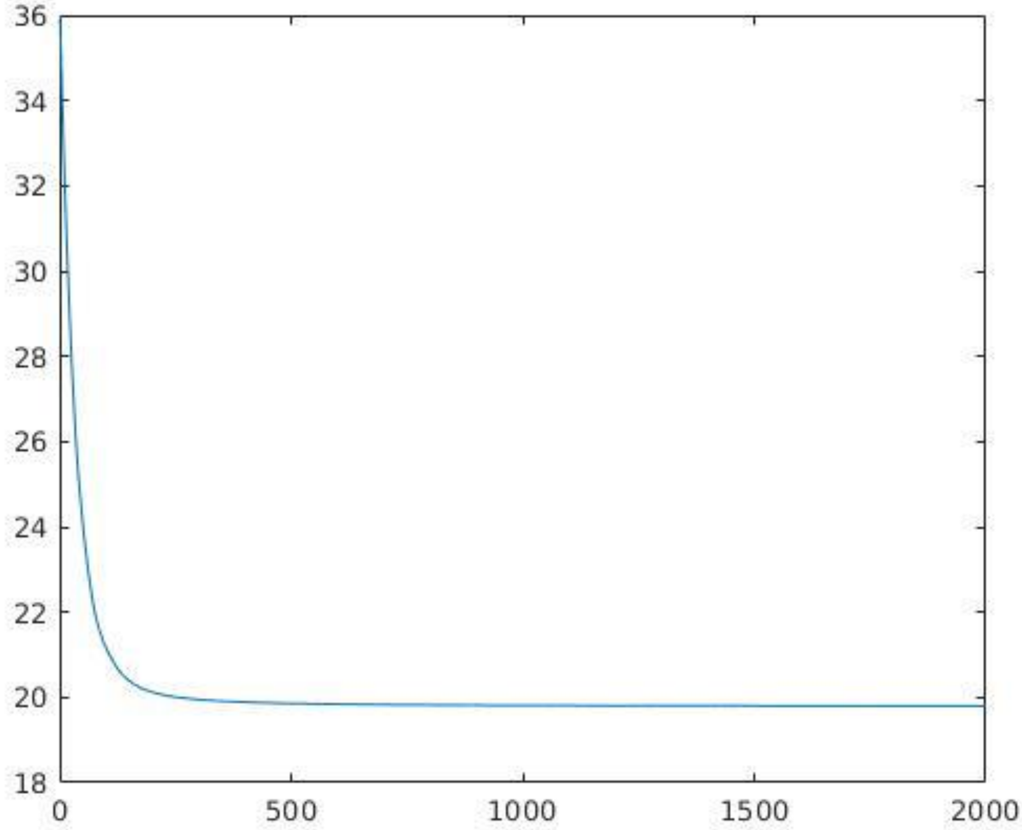
(3) Subgradient of L_i wrt. w_j for $j \neq y_i$ & $j \neq \hat{y}_i$:-

$$\frac{\partial L_i}{\partial w_j} = \frac{1}{n} w_j \quad \text{where } j \neq y_i \text{ \& } j \neq \hat{y}_i$$

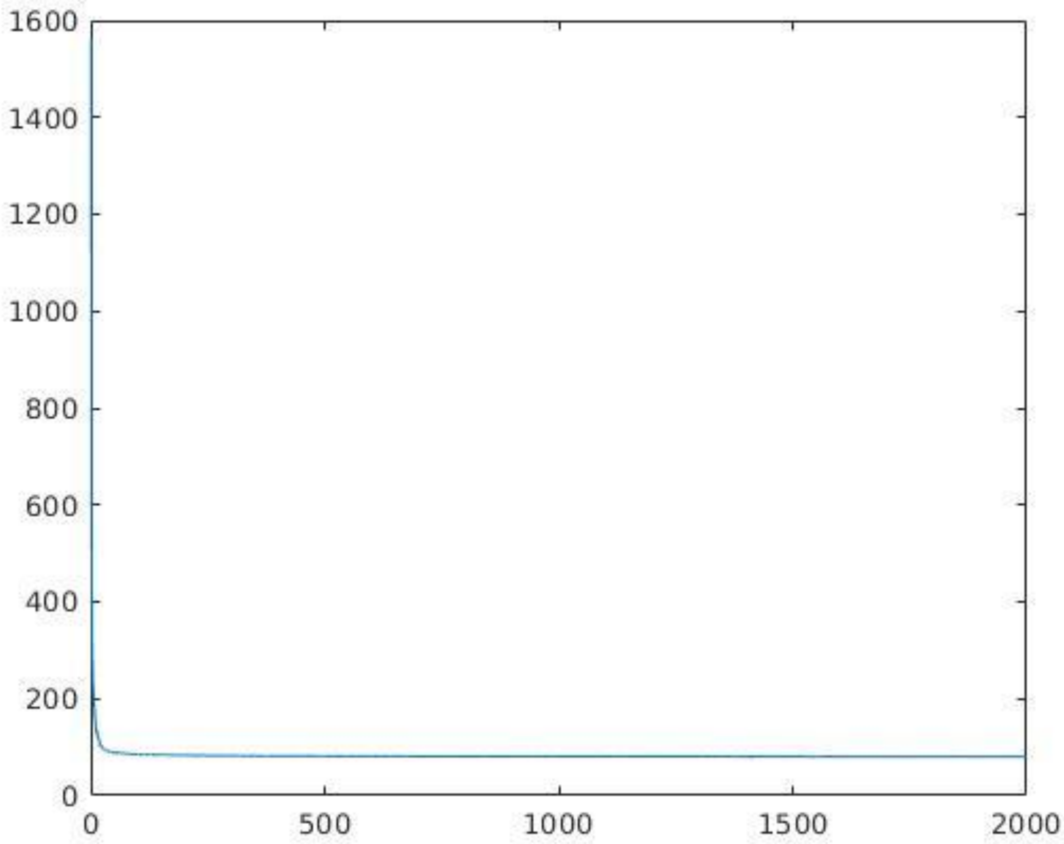
(3.2.5)

Using trD, trLb in q3 1 data.mat as your training set, run 2000 epochs over the dataset using η0 = 1, η1 = 100, C = 0.1 and C = 10. Plot the loss in Eq. (13) after each epoch. Compare with the objective value obtained in 3.1.4.

- For C = 0.1



- For C = 10



Objective Function Value:

For C = 10: 90

For C = 0.1: 22

(3.2.6) Using the W learned after 2000 epochs, report:

For C = 0.1

(a) Prediction error on valD, valLb, in q3_1_data.mat (test error) = 0.049

(b) The prediction error on trD, trLb (training error) = 0.0304

(c) 16.1101

For C = 10

(a) Prediction error on valD, valLb, in q3_1_data.mat (test error) = 0.0272

(b) The prediction error on trD, trLb (training error) = 0

(c) 120.95

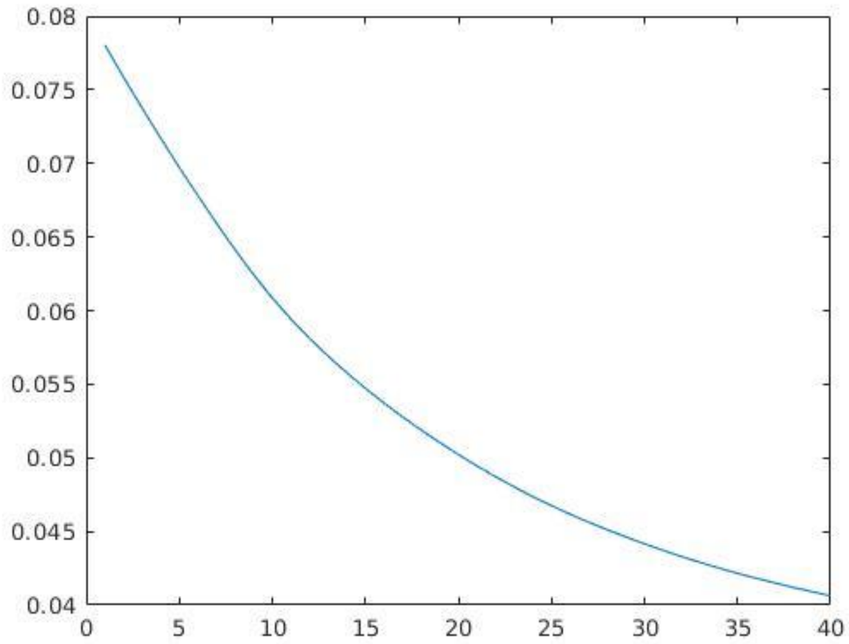
(3.2.7) Kaggle

1. **Best accuracy:** 0.81097

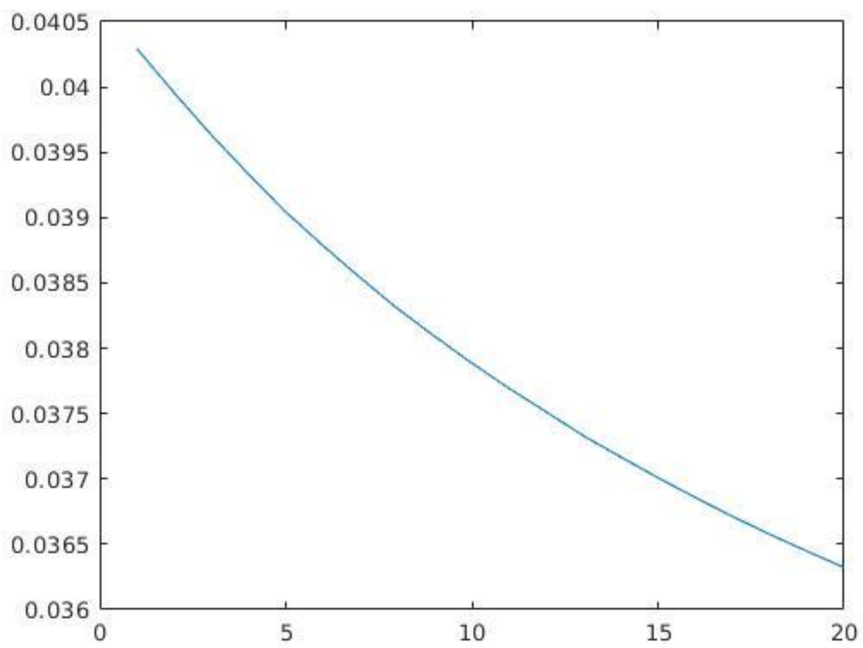
2. **Parameters:**
   a. C = 0.00001
   b. Epochs = 40 (Training Data)
   c. Epochs = 10 (Validation Data)

3. **Training Loss:**

   a. **Training Data**



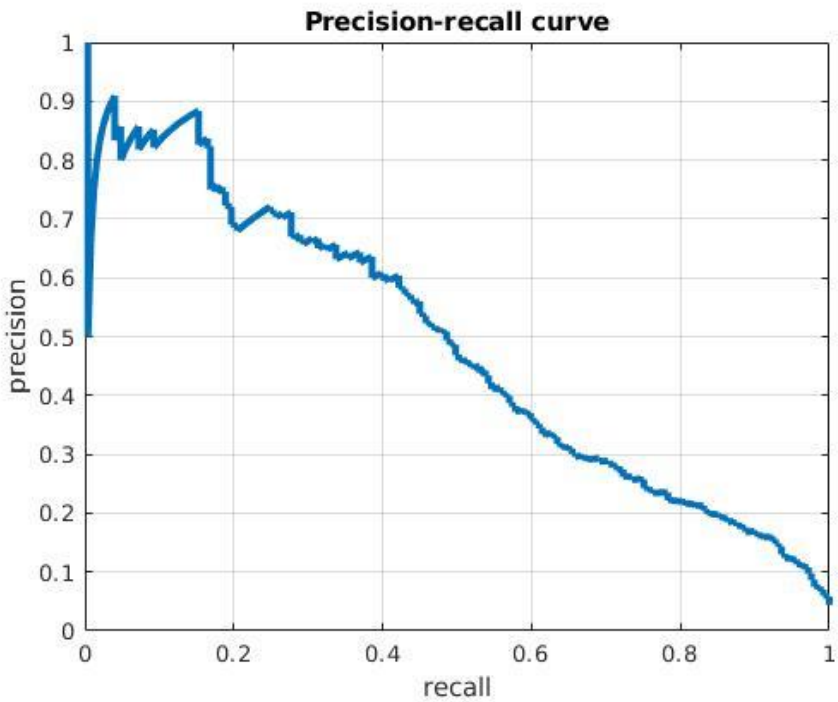
b. Validation Data



Q.4 SVM for object detection

(4.4.1) Use the training data in HW2 Utils.getPosAndRandomNeg() to train an SVM classifier. Use this classifier to generate a result file (use HW2 Utils.genRsltFile) for validation data. Use HW2 Utils.cmpAP to compute the AP and plot the precision recall curve. Submit your AP and precision recall curve (on validation data).

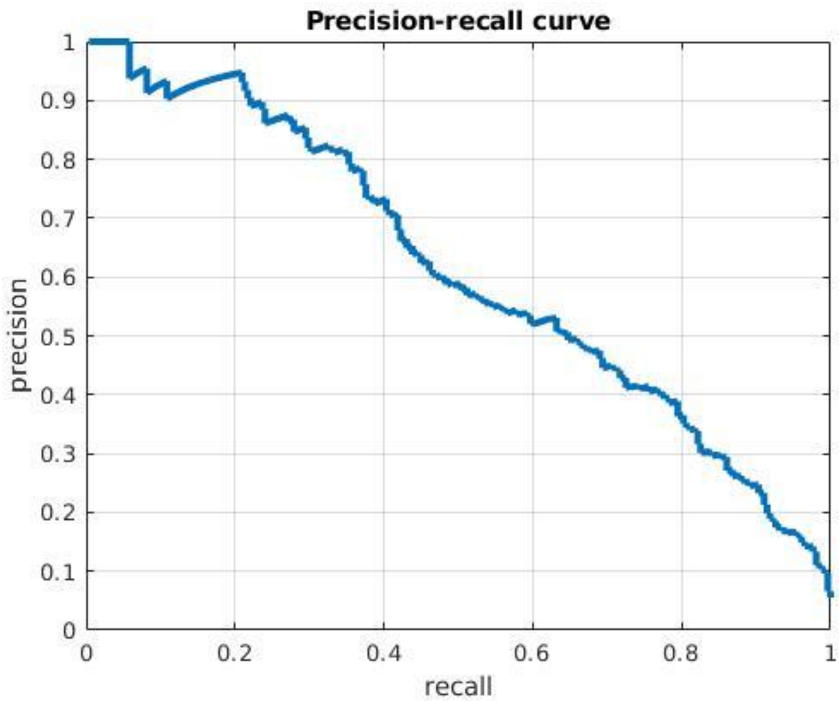
1. **For C = 10**
    a. Precision Recall Curve



b. ap = 0.4926

2. For C = 0.1

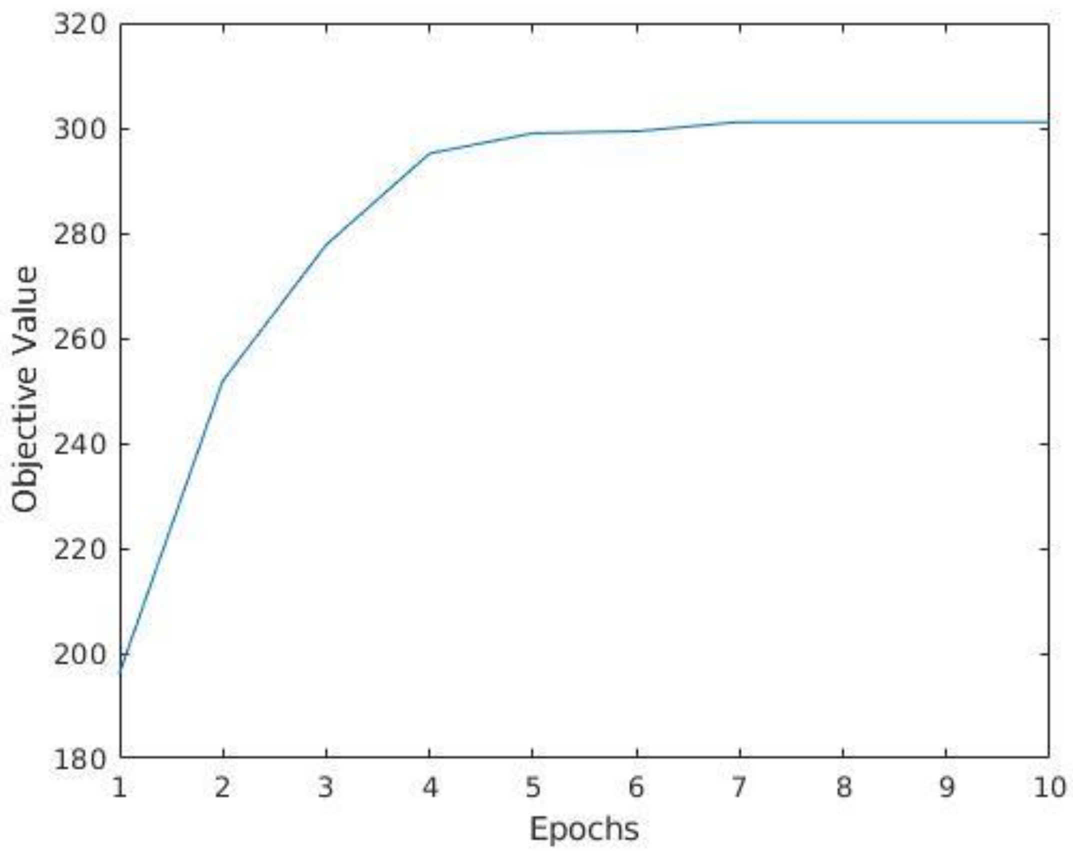
a. Precision Recall Curve



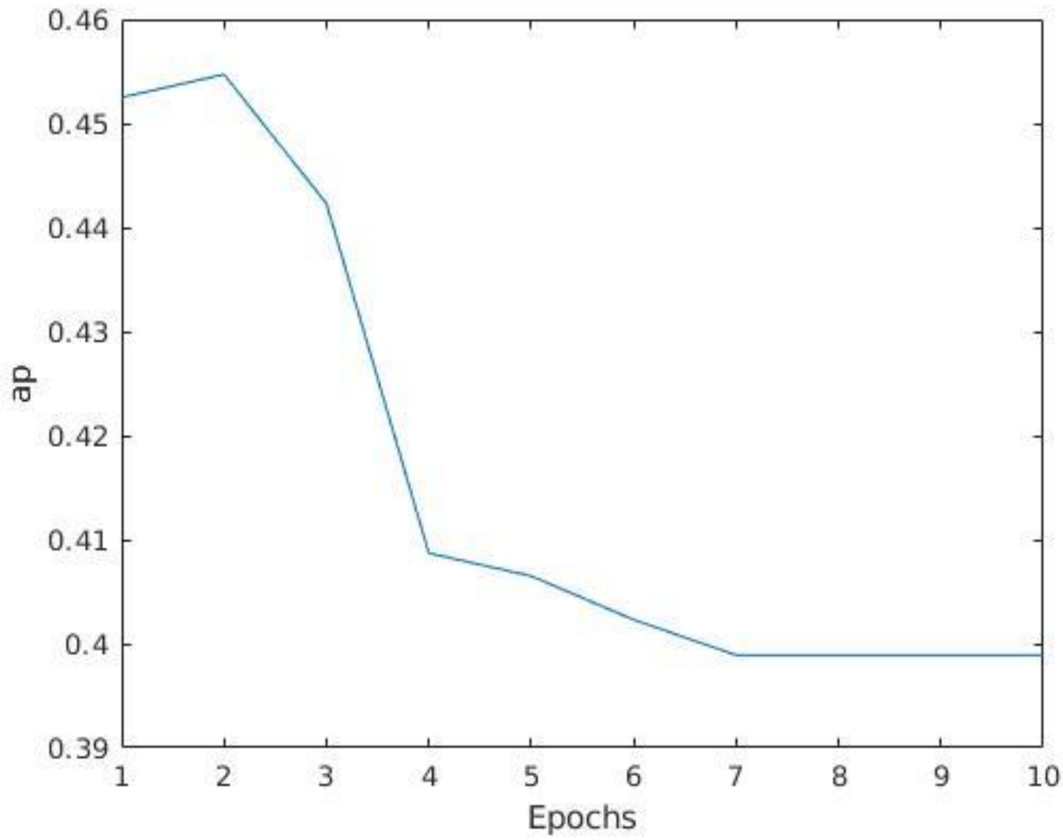
b. ap = 0.6164

(4.4.3) Run the negative mining for 10 iterations. Assume your computer is not so powerful and so you cannot add more than 1000 new negative training examples at each iteration. Record the objective values (on train data) and the APs (on validation data) through the iterations. Plot the objective values. Plot the APs.

1. **Objective**



2. **AP**



(4.4.4)

AP = 49.76