

# **CSCI-567: Assignment #4**

Due on Monday, November 9, 2015

**Saket Choudhary**  
**2170058637**

## Contents

<b>Problem 1</b>	<b>3</b>
Problem 1: (a) Gradient Calculation . . . . .	3
Problem 1: (b) Weak Learner Section . . . . .	3
Problem 1: (c) Step Size Selection . . . . .	4
<b>Problem 2</b>	<b>4</b>
Problem 2: (a) . . . . .	4
Problem 2: (b) . . . . .	5
Problem 2: (c) . . . . .	6
Problem 2: (d) . . . . .	7
<b>Problem 3.3</b>	<b>8</b>
Problem 3.3: (a) . . . . .	8
Problem 3.3: (b) . . . . .	8
Problem 3.3: (c) . . . . .	8
<b>Problem 3.4</b>	<b>8</b>
Problem 3.4: (a) . . . . .	8
Problem 3.4: (b) . . . . .	8
<b>Problem 3.5</b>	<b>8</b>
Problem 3.5: (a) . . . . .	8
Problem 3.5: (b) . . . . .	9

## Problem 1

### Problem 1: (a) Gradient Calculation

$$L(y_i, \hat{y}_i) = \log(1 + \exp(-y_i \hat{y}_i))$$

$$g_i = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}$$

$$g_i = \frac{-y_i \exp(-y_i \hat{y}_i)}{1 + \exp(-y_i \hat{y}_i)}$$

### Problem 1: (b) Weak Learner Section

$$h^* = \min_{h \in H} \left( \min_{\gamma \in R} \sum_{i=1}^n (-g_i - \gamma h(x_i))^2 \right)$$

$$\implies \frac{\partial h^*}{\partial \gamma} = 0$$

$$\implies 2 \sum_{i=1}^n (-g_i - \gamma h(x_i))(-h(x_i)) = 0$$

$$\hat{h} = -\frac{\sum_{i=1}^n g_i h(x_i)}{\sum_{i=1}^n h(x_i)^2}$$

Also check if it is indeed minimum with a second derivative test:

$$\frac{\partial^2 h^*}{\partial \gamma^2} = 2 \sum_{i=1}^n h(x_i)^2 > 0$$

Since the second derivative is positive definite,  $\hat{h}$  is indeed where the minima occurs.

**Problem 1: (c) Step Size Selection**

$$\alpha^* = \arg \min_{\alpha \in \mathbb{R}} \sum_{i=1}^n L(y_i, \hat{y}_i + \alpha h^*(x_i))$$

Newton's approximation:

$$\alpha_1 = \alpha_0 - \frac{f(\alpha_0)}{f'(\alpha_0)}$$

We start from  $\alpha_0 = 0$  and hence:

$$\begin{aligned} f(\alpha_0) &= \sum_{i=1}^n \log(1 + \exp(-y_i \hat{y}_i)) \\ f'(\alpha_0) &= \sum_{i=1}^n \frac{\partial L}{\partial \alpha} \\ &= \sum_{i=1}^n \frac{-y_i h^*(x_i) \exp(-y_i(\hat{y}_i + \alpha h^*(x_i)))}{1 + \exp(-y_i(\hat{y}_i + \alpha h^*(x_i)))} \\ &= \sum_{i=1}^n \frac{-y_i h^*(x_i) \exp(-y_i \hat{y}_i)}{1 + \exp(-y_i \hat{y}_i)} \end{aligned}$$

Thus,

$$\alpha_1 = \frac{\sum_{i=1}^n (\log(1 + \exp(-y_i \hat{y}_i)))}{\sum_{i=1}^n \frac{-y_i h^*(x_i) \exp(-y_i \hat{y}_i)}{1 + \exp(-y_i \hat{y}_i)}}$$

**Problem 2****Problem 2: (a)**

Primal form:

$$\begin{aligned} &\min_w ||w||^2 \\ &\text{such that } |y_i - (w^T x_i + b)| \leq \epsilon \end{aligned}$$

**Problem 2: (b)**

$$\min_{w, \epsilon_i} \frac{1}{2} \|w\|^2 + C \sum_i \epsilon_i$$

such that  $(w^T x_i + b) - y_i \leq n_i + \epsilon_i$  (positive deviation)

and  $y_i - (w^T x_i + b) \leq p_i + \epsilon_i$  (negative deviation)

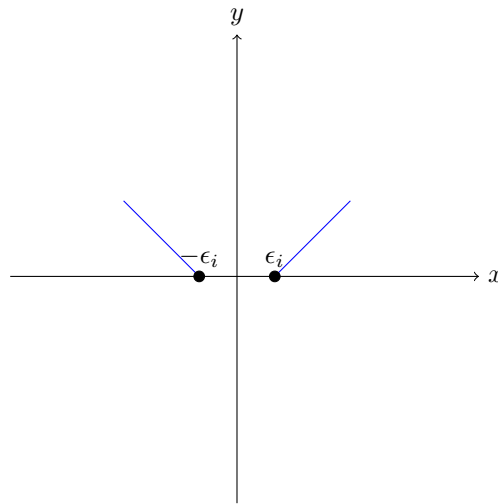
$$n_i \geq 0$$

$$p_i \geq 0$$

Also, the slackness loss needs further constraints:

$$n_i = \begin{cases} 0 & |n_i| < \epsilon_i, \\ |n_i| - \epsilon_i & \text{otherwise} \end{cases}$$

$$p_i = \begin{cases} 0 & |p_i| < \epsilon_i, \\ |p_i| - \epsilon_i & \text{otherwise} \end{cases}$$



So essentially  $n_i, p_i$  are non zero, only above the two blue lines

**Problem 2: (c)**

$$\begin{aligned}
L = & \frac{1}{2} \|w\|^2 + C \sum_i (p_i + n_i) \\
& - \sum_i (\eta_i p_i + \eta'_i n_i) \\
& - \sum_i \alpha_i (\epsilon + p_i - (y_i - (w^T x_i + b))) \\
& - \sum_i \beta_i (\epsilon + n_i + (y_i - (w^T x_i + b)))
\end{aligned}$$

**Conditions :**

$$\alpha_i \geq 0$$

$$\beta_i \geq 0$$

$$\eta_i \geq 0$$

$$\eta_i^* \geq 0$$

Dual Form:

$$\begin{aligned}
\Delta_w L &= 0 \\
&= w - \sum_i \alpha_i x_i + \sum_i \beta_i x_i = 0
\end{aligned}$$

$$\begin{aligned}
\Delta_b L &= 0 \\
&= \sum_i \alpha_i - \sum_i \beta_i = 0
\end{aligned}$$

$$\begin{aligned}
\Delta_{p_i} L &= 0 \\
&= C - \sum_i \eta_i - \sum_i \alpha_i = 0
\end{aligned}$$

$$\begin{aligned}
\Delta_{n_i} L &= 0 \\
&= C - \sum_i \eta'_i - \sum_i \beta_i = 0
\end{aligned}$$

Thus,  $w$  is given by:

$$w = \sum_i \alpha_i x_i - \sum_i \beta_i x_i$$

depends only on the support vectors.

This reduces the optimisation to:

$$\begin{aligned} \max f &= \frac{1}{2} \sum_{i,j} (\alpha_i - \beta_i) x_i^T x_j (\alpha_j - \beta_j) + p_i (C - \sum_i \eta_i - \sum_i \alpha_i) \\ &\quad + n_i (C + \sum_i \eta'_i - \sum_i \beta_i) \\ &\quad + \epsilon (-\sum_i \alpha_i - \sum_i \beta_i) \\ &\quad + \sum_i y_i (\alpha_i - \beta_i) \\ &= \frac{1}{2} \sum_{i,j} (\alpha_i - \beta_i) x_i^T x_j (\alpha_j - \beta_j) - \epsilon (\sum_i (\alpha_i + \beta_i)) \\ &\quad + \sum_i y_i (\alpha_i - \beta_i) \end{aligned}$$

such that  $\sum_i (\alpha_i - \beta_i) = 0$

and  $\alpha_i, \beta_i \in [0, C]$

## Problem 2: (d)

Using Kernel transformation:

$$w = \sum_i (\alpha_i - \beta_i) \phi(x_i)$$

this happens because  $x_i^T x_j$  gets mapped onto by an equivalent kernel function  $k(x_i, x_j) = \phi^T(x_i) \phi(x_j)$  and the objective function is:

$$\max_f = \frac{1}{2} \sum_{i,j} (\alpha_i - \beta_i) k(x_i, x_j) (\alpha_j - \beta_j) - \epsilon (\sum_i (\alpha_i + \beta_i))$$

## Problem 3.3

### Problem 3.3: (a)

C	Tr. Dataset 1(t)	Tr. Dataset 2(t)	Training Dataset 3(t)	CV
$4^{-6}=0.000244$	0.606156	0.400791	0.346078	0.578976
$4^{-5}=0.000977$	0.379495	0.477559	0.498054	0.907001
$4^{-4}=0.003906$	0.529940	0.495841	0.534946	0.926001
$4^{-3}=0.015625$	0.516236	0.577324	0.561874	0.935501
$4^{-2}=0.062500$	0.512287	0.529517	0.554510	0.945006
$4^{-1}=0.250000$	0.630195	0.663459	0.657651	0.943010
$4^0=1.000000$	0.746649	0.601710	0.563063	0.939003
$4^1=4.000000$	0.633370	0.572011	0.595552	0.942501
$4^2=16.000000$	0.674677	0.681010	0.698041	0.943503

As seen from the table. the time seems to increase with  $C$  and the  $CV$  increases too.  $C$  determines the tradeoff between objective function complexity and the overall loss. When  $C$  is small, there are chances of overfitting, this is evident from low  $CV$  values for lower  $C$  (because the generalisation error is high)

In terms of time complexity, if

The larger the value of  $C$ , the more is the penalisation and hence smaller the  $\epsilon_i$  would be

### Problem 3.3: (b)

Based on lowest cross validation error.  $C = 4^2$

### Problem 3.3: (c)

With  $C = 16$ , test accuracy = 0.943500

## Problem 3.4

### Problem 3.4: (a)

'libsvm' gives 0.9455 as its accuracy which is pretty close to

### Problem 3.4: (b)

'libsvm' gives 0.9455 as its accuracy which is pretty close to

## Problem 3.5

### Problem 3.5: (a)



**Problem 3.5: (b)**

--