

CSCI-567: Assignment #4

Due on Monday, November 9, 2015

Saket Choudhary
2170058637

Contents

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

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

And,

$$\begin{aligned} f''(\alpha_0) &= \sum_{i=1}^n \frac{\partial^2 L}{\partial \alpha^2} \\ &= \sum_{i=1}^n \frac{\{(1 + \exp(-y_i(\hat{y}_i + \alpha h^*(x_i))))(y_i h^*(x_i))^2 + y_i h^*(x_i)\} \exp(-y_i(\hat{y}_i + \alpha h^*(x_i)))}{(1 + \exp(-y_i(\hat{y}_i + \alpha h^*(x_i))))^2} \\ &= \sum_{i=1}^n \frac{\{(1 + \exp(-y_i \hat{y}_i))(y_i h^*(x_i))^2 + y_i h^*(x_i)\} \exp(-y_i \hat{y}_i)}{(1 + \exp(-y_i \hat{y}_i))^2} \end{aligned}$$

Thus,

$$\alpha_1 = \frac{\sum_{i=1}^n \frac{y_i h^*(x_i) \exp(-y_i \hat{y}_i)}{1 + \exp(-y_i \hat{y}_i)}}{\sum_{i=1}^n \frac{\{(1 + \exp(-y_i \hat{y}_i))(y_i h^*(x_i))^2 + y_i h^*(x_i)\} \exp(-y_i \hat{y}_i)}{(1 + \exp(-y_i \hat{y}_i))^2}}$$

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)

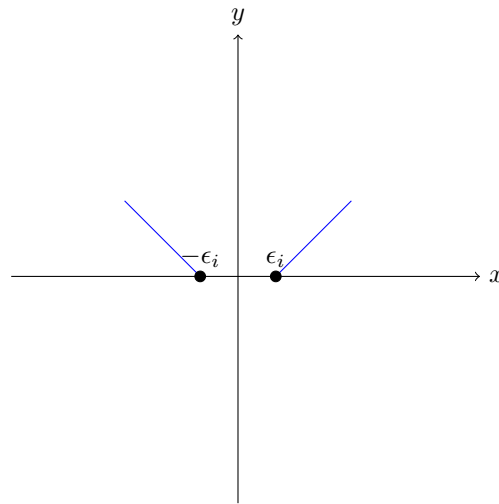
$$n_i \geq 0$$

$$p_i \geq 0$$

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:

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

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

$$\text{and } \alpha_i, \beta_i \in [0, C]$$

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	Avg. t
$4^{-6}=0.000244$	0.606156	0.400791	0.346078	0.578976	0.451
$4^{-5}=0.000977$	0.379495	0.477559	0.498054	0.907001	0.451
$4^{-4}=0.003906$	0.529940	0.495841	0.534946	0.926001	0.520
$4^{-3}=0.015625$	0.516236	0.577324	0.561874	0.935501	0.551
$4^{-2}=0.062500$	0.512287	0.529517	0.554510	0.945006	0.532
$4^{-1}=0.250000$	0.630195	0.663459	0.657651	0.943010	0.650
$4^0=1.000000$	0.746649	0.601710	0.563063	0.939003	0.637
$4^1=4.000000$	0.633370	0.572011	0.595552	0.942501	0.600
$4^2=16.000000$	0.674677	0.681010	0.698041	0.943503	0.684

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 ϵ_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)

Platform Used: <i>Ubuntu</i> 12.04, <i>x86_64</i>		
'libsvm' gives 0.9655 as its accuracy which is pretty close to 0.943 that my code gives.		
C	Training Time	CV
4^{-6}	0.829624	0.5575
4^{-5}	0.827846	0.5575
4^{-4}	0.831586	0.5575
4^{-3}	0.831599	0.7295
4^{-2}	0.649459	0.9195
4^{-1}	0.425499	0.934
4^0	0.281219	0.949
4^1	0.210071	0.955
4^2	0.201989	0.9655

Problem 3.4: (b)

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

Problem 3.5**Problem 3.5: (a)**

C	Training Time	CV
0.015625	0.675134	0.7010
0.062500	0.680862	0.5575
0.250000	0.682373	0.5575
1.000000	0.532627	0.9175
4.000000	0.608079	0.8655
16.000000	0.678589	0.7690
64.000000	0.347019	0.9280
256.000000	0.432110	0.9255
1024.000000	0.523863	0.9180
4096.000000	0.239251	0.9385
16384.000000	0.284755	0.9470

Problem 3.5: (b)

C	Training Time	CV
0.015625	0.806863	0.5575
0.062500	0.804094	0.5575
0.250000	0.804537	0.5575
1.000000	0.807285	0.5575
4.000000	0.810119	0.6475
16.000000	0.815702	0.7375
64.000000	0.828215	0.55750
256.000000	0.803329	0.5575
1024.000000	0.803978	0.5575
4096.000000	0.806347	0.5575
16384.000000	0.816474	0.8335

Based on the RBF, polykernel:

Polynomial Kernel train accuracy: 96.600000

RBF Kernel train accuracy: 96.900000

Better kernel: RBF

Polynomial Kernel optimal C: 64.000000

Polynomial Kernel optimal degree: 2.000000

Polynomial Kernel test accuracy: 95.150000

RBF Kernel optimal g: 0.062500

RBF Kernel optimal c: 16.000000

RBF Kernel test accuracy: 96.500000