CS480/680: Introduction to Machine Learning

Homework 1

Due: 11:59 pm, May 29, 2024, submit on LEARN.

Prajwal Thakue

21052875

Submit your writeup in pdf and all source code in a zip file (with proper documentation). Write a script for each programming exercise so that the TA can easily run and verify your results. Make sure your code runs!
[Text in square brackets are hints that can be ignored.]

Exercise 1: Perceptron (8 pts)

Convention: All algebraic operations, when applied to a vector or matrix, are understood to be element-wise (unless otherwise stated).

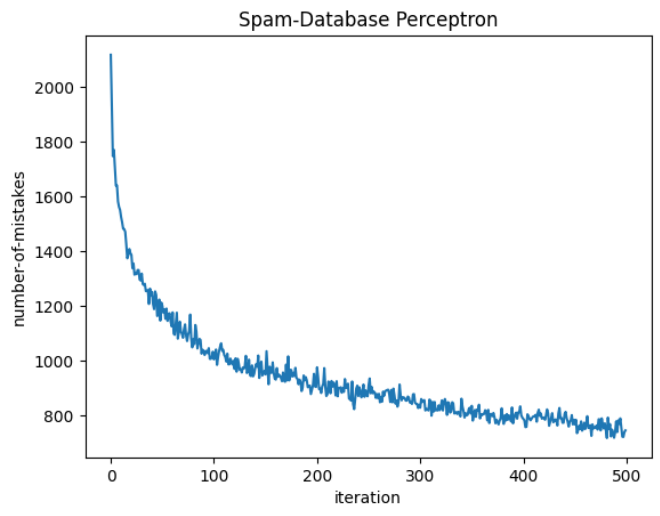
Algorithm 1: The perceptron.

Input: $X \in \mathbb{R}^{d \times n}$, $\mathbf{y} \in \{-1, 1\}^n$, $\mathbf{w} = \mathbf{0}_d$, $b = 0$, $\mathsf{max\_pass} \in \mathbb{N}$
Output: $\mathbf{w}, b, mistake$

```
for t = 1, 2, ..., max_pass do
    mistake(t) ← 0
    for i = 1, 2, ..., n do
        if y_i(⟨x_i, w⟩ + b) ≤ 0 then
            w ← w + y_i x_i                    // x_i is the i-th column of X
            b ← b + y_i
            mistake(t) ← mistake(t) + 1
```

1. (1 pt) Implement the perceptron in Algorithm 1. Your implementation should take input as $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, $\mathbf{y} \in \{-1, 1\}^n$, an initialization of the hyperplane parameters $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, and the maximum number of passes of the training set [suggested $\mathsf{max\_pass} = 500$]. Run your perceptron algorithm on the spambase dataset (available on course website), and plot the number of mistakes (y-axis) w.r.t. the number of passes (x-axis).

   Ans:



2. (1 pt) Using the one-vs-all reduction to implement a multiclass perceptron. You may call your binary implementation. Test your algorithm on the activity dataset (available on course website), and report your final errors on the training and test sets.

   Ans: Number of mistakes in training (last iter) for each model: [0. 0. 0. 40. 38. 0.] Number of mistakes in training (last iter) sum: 78 Number of mistakes in test: 145

3. (1 pt) Using the one-vs-one reduction to implement a multiclass perceptron. You may call your binary implementation. Test your algorithm on the activity dataset (available on course website), and report your final errors on the training and test sets.

   Ans: Number of mistakes in training (last iter) for each model: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 44. 0. 0.]
   Number of mistakes in total-training-error= 44.0
   Total mistakes-in-test-data 148

4. (2 pts) Consider the (continuous) piece-wise function

$$f(\mathbf{w}) := \max_k f_k(\mathbf{w}), \tag{1}$$

   where each f_k is continuously differentiable. We define the derivative of f at any $\mathbf{w}$ as follows: first find (any) k such that $f(\mathbf{w}) = f_k(\mathbf{w})$, i.e., $f_k(\mathbf{w})$ achieves the maximum among all pieces; then we let $f'(\mathbf{w}) = f'_k(\mathbf{w})$. [Clearly, the index k that achieves maximum may depend on $\mathbf{w}$, the point we evaluate the derivative at.] Now consider the following problem [padding applied, $y_i \in \{\pm 1\}$]:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \max\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle)\}. \tag{2}$$

   Prove that in each iteration, the (binary) perceptron algorithm essentially picks a term from the above summation, computes the corresponding derivative (say $\mathbf{g}$), and performs a gradient update:

$$\mathbf{w} \leftarrow \mathbf{w} - \mathbf{g}. \tag{3}$$

   [You may ignore the degenerate case when $\langle \mathbf{x}_i, \mathbf{w}\rangle = 0$, and you can use the usual chain rule for our derivative.]

   Ans: Question-4: For any training-sample i, we have following

$$\nabla_w \max\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle)\} = \begin{cases} 0, & \text{if } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) > 0 \because \max\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) = 0 \text{ and } \nabla_w 0 = 0 \\ -y_i\mathbf{x}_i, & \text{if } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) \leq 0 \because \max\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) = -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle \text{ and} \\ & -\nabla_w y_i \langle \mathbf{x}_i, \mathbf{w}\rangle = -\nabla_w y_i \mathbf{x}_i^T \mathbf{w} = -y_i \mathbf{x}_i \end{cases} \tag{4}$$

   So , We pick any term (i.e a training sample i)from the summation ($\sum_{i=1}^{n} \max\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle)\}$) , The corresponding gradient is given by eqn-(4) . The gradient update then would be :

$$\begin{cases} w \leftarrow w - 0, & \text{if } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) > 0 \text{ i.e no update if no mistake} \\ w \leftarrow w - (-y_i\mathbf{x}_i) = w \leftarrow w + y_i\mathbf{x}_i, & \text{if } y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle) \leq 0 \text{ i.e update if make a mistake} \end{cases} \tag{5}$$

   Gradient update shown in equation-(5) is exactly same as in shown in algo-(1) , Note: Given that we have Already Applied padding therefore the equation-(5) will update the bias term b as well.
   ■

5. (1 pt) Consider the following problem, where $y_i \in \{1, 2, \ldots, c\}$:

$$\min_{\mathbf{w}_1, \ldots, \mathbf{w}_c} \sum_{i=1}^{n} \max_{k=1,\ldots,c} \Big[\langle \mathbf{x}_i, \mathbf{w}_k\rangle - \langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle\Big]. \tag{6}$$

   Show that when $c = 2$, we reduce to the binary perceptron problem in (2). [Try to identify the weights $\mathbf{w}$, using some transformation.]

   Ans: Question-5: for c=2: We relabel the classes as $y_i \in \{+1, -1\}$

$$\min_{\mathbf{w}_{+1}, \mathbf{w}_{-1}} \sum_{i=1}^{n} \max_{k=+1,-1} \Big[\langle \mathbf{x}_i, \mathbf{w}_k\rangle - \langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle\Big]. \tag{7}$$

$$\max_{k=+1,-1} \Big[\langle \mathbf{x}_i, \mathbf{w}_k\rangle - \langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle\Big] = \begin{cases} \max\Big\{0, \langle \mathbf{x}_i, \mathbf{w}_{-1} - \mathbf{w}_{+1}\rangle\Big\}, & \text{if } y_i = +1 \\ \max\Big\{0, \langle \mathbf{x}_i, \mathbf{w}_{+1} - \mathbf{w}_{-1}\rangle\Big\}, & \text{if } y_i = -1 \end{cases} \tag{8}$$

equation-(8) , can be written as :

$$\max\left\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}_{+1} - \mathbf{w}_{-1}\rangle)\right\} \tag{9}$$

$$\text{since } \max\left\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}_{+1} - \mathbf{w}_{-1}\rangle)\right\} = \begin{cases} \max\left\{0, \langle \mathbf{x}_i, \mathbf{w}_{-1} - \mathbf{w}_{+1}\rangle\right\}, & \text{if } y_i = +1 \\ \max\left\{0, \langle \mathbf{x}_i, \mathbf{w}_{+1} - \mathbf{w}_{-1}\rangle\right\}, & \text{if } y_i = -1 \end{cases} \tag{10}$$

rewriting the Equation-(9):

$$\text{let define, } \mathbf{v} \triangleq \begin{bmatrix} \mathbb{I}_{d+1\times d+1} \\ -\mathbb{I}_{d+1\times d+1} \end{bmatrix} \text{ where } d \textbf{ num of features + 1(padding)} \tag{11}$$

$$\text{Then we can define } \mathbf{w} \triangleq \mathbf{v}^T \begin{bmatrix} \mathbf{w}_{+1} \\ \mathbf{w}_{-1} \end{bmatrix} \tag{12}$$

$$\text{Thus,} \tag{13}$$

$$\max_{k=+1,-1}\left[\langle \mathbf{x}_i, \mathbf{w}_k\rangle - \langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle\right] = \max\left\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}_{+1} - \mathbf{w}_{-1}\rangle)\right\} = \max\left\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle)\right\} \tag{14}$$

Thus,

$$\min_{\mathbf{w}_{+1}, \mathbf{w}_{-1}} \sum_{i=1}^{n} \max_{k=+1,-1}\left[\langle \mathbf{x}_i, \mathbf{w}_k\rangle - \langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle\right] = \min_{\mathbf{w}} \sum_{i=1}^{n} \max\left\{0, -y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle)\right\} \tag{15}$$

■

6. (2 pts) Based on the analogy to the binary case, develop and implement a multiclass perceptron algorithm to solve (6) directly. Run your implementation on the activity dataset (available on course website) and report the final errors on the training and test sets. [Hint: obviously, we want to predict as follows: $\hat{y} = \operatorname{argmax}_{k=1,\ldots,c} \langle \mathbf{x}, \mathbf{w}_k\rangle$, i.e., the class k whose corresponding $\mathbf{w}_k$ maximizes the inner product. Explain your algorithm (e.g., through pseudo-code).]

Ans: Following is the Pseudo Code

Algorithm 2: The MultiClass Perceptron

Input: $X \in \mathbb{R}^{d\times n}$, $\mathbf{y} \in \{-1, 1\}^n$, $\mathbf{w} = \mathbf{0}_d$, $b = 0$, max_pass $\in \mathbb{N}$
Output: $\mathbf{w}, b, mistake$

1 **for** $t = 1, 2, \ldots,$ max_pass **do**
2 $mistake(t) \leftarrow 0$
3 **for** $i = 1, 2, \ldots, n$ **do**
4 $\hat{y}_i = \operatorname{argmax}_{k\in 1,2\ldots K}(\langle \mathbf{x}_i, \mathbf{w}_k\rangle + b_k - (\langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle + b_{y_i}))$
5 **if** $y_i \neq \hat{y}_i$ **then**
6 $\mathbf{w}_{y_i} \leftarrow \mathbf{w}_{y_i} + \mathbf{x}_i$
7 $b_{y_i} \leftarrow b_{y_i} + 1$
8 $\mathbf{w}_{\hat{y}_i} \leftarrow \mathbf{w}_{\hat{y}_i} - \mathbf{x}_i$
9 $b_{\hat{y}_i} \leftarrow b_{\hat{y}_i} - 1$
10 $mistake(t) \leftarrow mistake(t) + 1$

The line 6-9 , represents gradient update as described below: Let, $\hat{y}_j = \operatorname{argmax}_{k\in 1,2\ldots K}(\langle \mathbf{x}_i, \mathbf{w}_k\rangle + b - (\langle \mathbf{x}_i, \mathbf{w}_{y_i}\rangle + b_{y_i}))$

Then, Let, $L = (\langle \mathbf{x}_i, \mathbf{w}_{y_i} \rangle + b_{y_i} - (\langle \mathbf{x}_i, \mathbf{w}_{y_i} \rangle + b_{y_i})$

$$w_{\hat{y}_i} \leftarrow w_{\hat{y}_i} - \frac{\partial L}{\partial w_k}\Big|_{\hat{y}_i} \qquad \frac{\partial L}{\partial w_k}\Big|_{\hat{y}_i} = +\mathbf{x}_i \tag{16}$$

$$w_{y_i} \leftarrow w_{y_i} - \frac{\partial L}{\partial w_{y_i}}\Big|_{y_i} \qquad \frac{\partial L}{\partial w_{y_i}}\Big|_{y_i} = -\mathbf{x}_i \tag{17}$$

$$b_{\hat{y}_i} \leftarrow b_{\hat{y}_i} - \frac{\partial L}{\partial b_k}\Big|_{\hat{y}_i} \qquad \frac{\partial L}{\partial b_{\hat{y}_i}}\Big|_{\hat{y}_i} = +1 \tag{18}$$

$$b_{y_i} \leftarrow b_{y_i} - \frac{\partial L}{\partial b_{y_i}}\Big|_{y_i} \qquad \frac{\partial L}{\partial b_{y_i}}\Big|_{y_i} = -1 \tag{19}$$

$$\tag{20}$$

Number of mistakes in training (last iter): 40
Number of mistakes in test: 132

Exercise 2: Generalized linear models (6 pts)

Recall that in logistic regression we assumed the *binary* label $\mathsf{Y}_i \in \{0, 1\}$ follows the Bernoulli distribution: $\Pr(\mathsf{Y}_i = 1|\mathsf{X}_i) = p_i$, where p_i also happens to be the mean. Under the independence assumption we derived the (conditional) negative log-likelihood function:

$$-\sum_{i=1}^{n}(1 - y_i)\log(1 - p_i) + y_i \log(p_i). \tag{21}$$

Then, we parameterized the mean parameter p_i through the logit transform:

$$\log \frac{p_i}{1 - p_i} = \langle \mathbf{x}_i, \mathbf{w} \rangle + b, \quad \text{or equivalently} \quad p_i = \frac{1}{1 + \exp(-\langle \mathbf{x}_i, \mathbf{w} \rangle - b)}. \tag{22}$$

Lastly, we found the weight vector $\mathbf{w}$ and b by minimizing the negative log-likelihood function.

In the following we generalize the above idea significantly. Let the (conditional) density of Y (given $\mathsf{X} = \mathbf{x}$) be

$$p(y|\mathbf{x}) = \exp\Big[\mu(\mathbf{x}) \cdot y - \lambda(\mathbf{x})\Big] \cdot q(y), \tag{23}$$

where $\mu : \mathbb{R}^d \to \mathbb{R}$ is a function of $\mathbf{x}$ and $\lambda(\mathbf{x}) = \log \int_y \exp\big(\mu(\mathbf{x}) \cdot y\big) q(y) \mathrm{d}y$ so that $p(y|\mathbf{x})$ is properly normalized wrt y (i.e., integrate to 1). For discrete y (such as in logistic regression), replace the density with the probability mass function and the integral with sum.

As always, you need to supply sufficient derivation details to justify your final answer.

1. (1 pt) Given a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, derive the (conditional) negative log-likelihood function of $y_1, \ldots, y_n$, assuming independence and the density form in (23).

   Ans: ref:stanford CS229 (https://www.youtube.com/watch?v=sj0iPn03i7Q)

$$Pr(Y_1 = y_1, Y_2 = y_2 \ldots Y_n = y_n | X_1 = x_1, X_2 = x_2 \ldots X_n = x_n) = \prod_{i=1}^{n} Pr(Y_i = y_i | X_i = x_i) \tag{24}$$

$$\text{From eqn (23)} \tag{25}$$

$$= \prod_{i=1}^{n} \exp\Big[\mu(\mathbf{x}_i) \cdot y_i - \lambda(\mathbf{x}_i)\Big] \cdot q(y_i) \tag{26}$$

$$\text{(conditional) negative log–likelihood of (26) is} \tag{27}$$

$$= -\sum_{i=1}^{i=n} log\left[\exp\Big[\mu(\mathbf{x}_i) \cdot y_i - \lambda(\mathbf{x}_i)\Big] \cdot q(y_i)\right] \tag{28}$$

2. (1 pt) Plug the usual linear parameterization

$$\mu(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b = \langle \mathbf{x}, \mathbf{w} \rangle \tag{29}$$

into your (conditional) negative log-likelihood and compute the gradient of the resulting function. [Hint: you may swap differentiation with integral and your gradient may involve implicitly defined terms.]

Ans: Plug-in the definition of $\mu(\mathbf{x})$ and $\lambda(\mathbf{x})$ in equation (28) , we get

$$\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} log\left[\exp\left[\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \log \int_{y_i} \exp\big(\mu(\mathbf{x}_i) \cdot y_i\big) q(y_i) \mathrm{d}y_i\right] \cdot q(y_i)\right] \tag{30}$$

$$= -\sum_{i=1}^{i=n} \left[\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \log \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i + \log(q(y_i))\right] \tag{31}$$

and hence

$$\nabla \ell_n(\mathbf{w}) = -\nabla_w \sum_{i=1}^{i=n} \left[\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \log \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i + \log(q(y_i))\right] \tag{32}$$

$$= -\sum_{i=1}^{i=n} \nabla \left[\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \log \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i + \log(q(y_i))\right] \tag{33}$$

$$= -\sum_{i=1}^{i=n} \left[\nabla(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i) - \nabla\Big(\log \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i\Big)\right] \qquad \because \nabla_{\mathbf{w}}(\log(q(y_i))) = 0 \tag{34}$$

$$= -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \frac{\nabla \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i}{\int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i}\right] \tag{35}$$

$$= \nabla \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i = \int_{y_i} \nabla\big(exp(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i) q(y_i) \mathrm{d}y_i\big) \tag{36}$$

$$= \mathbf{x}_i \int_{y_i} y_i \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i \tag{37}$$

$$\therefore \nabla \ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \frac{\mathbf{x}_i \int_{y_i} y_i \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i}{\int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i}\right] \tag{38}$$

$$\because exp(\lambda(\mathbf{x}_i)) = \int_{y_i} \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i \tag{39}$$

$$\nabla \ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \frac{\mathbf{x}_i \int_{y_i} y_i \exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i\big) q(y_i) \mathrm{d}y_i}{exp(\lambda(\mathbf{x}_i))}\right] \tag{40}$$

$$= -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \mathbf{x}_i \int_{y_i} y_i \underbrace{\exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \lambda(\mathbf{x}_i)\big) q(y_i)}_{P(y_i|\mathbf{x}_i)} \mathrm{d}y_i\right] \tag{41}$$

$$= -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \mathbf{x}_i \int_{y_i} y_i \underbrace{\exp\big(\langle \mathbf{x}_i, \mathbf{w} \rangle \cdot y_i - \lambda(\mathbf{x}_i)\big) q(y_i)}_{P(y_i|\mathbf{x}_i)} \mathrm{d}y_i\right] \tag{42}$$

$$\because \mathbf{E}(y_i|\mathbf{x}_i; \mathbf{w}) = \int_{y_i} y_i \mathbf{P}(y_i|\mathbf{x}_i) \mathrm{d}y_i \tag{43}$$

$$\nabla \ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \mathbf{x}_i \cdot \mathbf{E}(y_i|\mathbf{x}_i; \mathbf{w})\right] \tag{44}$$

3. (1 pt) Let us revisit linear regression, where

$$p(y|\mathbf{x}) = \tfrac{1}{\sqrt{2\pi}} \exp\big(-\tfrac{(y-\nu(\mathbf{x}))^2}{2}\big) \tag{45}$$

Identify the functions $\mu(\mathbf{x})$, $\lambda(\mathbf{x})$ and $q(y)$ for the above specialization. Based on the linear parameterization in Ex 2.2, derive the negative log-likelihood and gradient. [Hint: you may simply plug into the more general result in Ex 2.2. Compare with what you already learned about linear regression to make sure both Ex 2.2 and Ex 2.3 are correct.]

Ans: We have ,

$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-\nu(\mathbf{x}))^2}{2}\right) \tag{46}$$

Expanding equation-(46) and rewriting in the expression similar to equation-(23)

$$p(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \exp\left(\nu(\mathbf{x})y - \frac{\nu(\mathbf{x})^2}{2}\right) \tag{47}$$

$$\mu(\mathbf{x}) = \nu(\mathbf{x}) \tag{48}$$

$$\lambda(\mathbf{x}) = \frac{\nu(\mathbf{x})^2}{2} \tag{49}$$

$$q(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \tag{50}$$

using (28) (51)

$$\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} \left[\mu(\mathbf{x}_i) \cdot y_i - \lambda(\mathbf{x}_i) + \log q(y_i)\right] \tag{52}$$

$$\because \mu(\mathbf{x}) = \langle \mathbf{x}_i, \mathbf{w}\rangle, \therefore \lambda(\mathbf{x}) = \frac{\langle \mathbf{x}_i, \mathbf{w}\rangle^2}{2} \tag{53}$$

$$= -\sum_{i=1}^{i=n} \left[\langle \mathbf{x}_i, \mathbf{w}\rangle \cdot y_i - \frac{\langle \mathbf{x}_i, \mathbf{w}\rangle^2}{2} + \log\frac{1}{\sqrt{2\pi}} - y_i\right] \tag{54}$$

$$= -\sum_{i=1}^{i=n} \left[\langle \mathbf{x}_i, \mathbf{w}\rangle \cdot y_i - \frac{\langle \mathbf{x}_i, \mathbf{w}\rangle^2}{2} + \log\frac{1}{\sqrt{2\pi}} - y_i\right] \tag{55}$$

Taking Gradient of equation-(55) wrt to $\mathbf{w}$ (56)

$$\nabla \ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n} \left[\mathbf{x}_i \cdot y_i - \mathbf{x}_i \cdot \langle \mathbf{x}_i, \mathbf{w}\rangle\right] \tag{57}$$

The expression (57) can also be found by directly substituting $\mathbf{E}(y_i|\mathbf{x}_i; \mathbf{w})$ (58)

by the value of $\nu(\mathbf{x})$, which in this case is equal to $\nu(\mathbf{x}) = \mu(\mathbf{x}) = \langle \mathbf{x}_i, \mathbf{w}\rangle$ (59)

(60)

4. (1 pt) Let us revisit logistic regression, where

$$\Pr(\mathsf{Y} = y|\mathbf{x}) = [\nu(\mathbf{x})]^y [1-\nu(\mathbf{x})]^{1-y}, \quad \text{where} \quad y \in \{0, 1\}. \tag{61}$$

Identify the functions $\mu(\mathbf{x})$, $\lambda(\mathbf{x})$ and $q(y)$ for the above specialization. Based on the linear parameterization in Ex 2.2, derive the negative log-likelihood and gradient. [Hint: Compare with what you already learned about logistic regression.]

Ans: We have ,

$$\Pr(\mathsf{Y} = y|\mathbf{x}) = [\nu(\mathbf{x})]^y [1-\nu(\mathbf{x})]^{1-y} \tag{62}$$

can re-write the equation-(62) as

$$= \log\left[exp\Big([\nu(\mathbf{x})]^y[1-\nu(\mathbf{x})]^{1-y}\Big)\right] \tag{63}$$

$$= \exp\left[log\Big([\nu(\mathbf{x})]^y[1-\nu(\mathbf{x})]^{1-y}\Big)\right] \tag{64}$$

$$= \log\left[exp\Big([\nu(\mathbf{x})]^y[1-\nu(\mathbf{x})]^{1-y}\Big)\right] \tag{65}$$

$$= \exp\left[\log\Big([\nu(\mathbf{x})]^y\Big) + \log\Big([1-\nu(\mathbf{x})]^{1-y}\Big)\right] \tag{66}$$

$$= \exp\left[y\cdot\log\Big([\nu(\mathbf{x})]\Big) + (1-y)\cdot\log\Big([1-\nu(\mathbf{x})]\Big)\right] \tag{67}$$

$$= \exp\left[\log\Big(\tfrac{[\nu(\mathbf{x})]}{[1-\nu(\mathbf{x})]}\Big)\cdot y + (1)\cdot\log\Big([1-\nu(\mathbf{x})]\Big)\right] \tag{68}$$

Comparing (68) with (23), we get

$$\mu(\mathbf{x}) = \log\Big(\tfrac{[\nu(\mathbf{x})]}{[1-\nu(\mathbf{x})]}\Big) \tag{69}$$

$$\nu(\mathbf{x}) = \tfrac{1}{1+exp(-\mu(\mathbf{x}))} \tag{70}$$

$$\lambda(\mathbf{x}) = -1\cdot\log\Big([1-\nu(\mathbf{x})]\Big) \tag{71}$$

$$= \log\Big([\tfrac{1}{1-\nu(\mathbf{x})}]\Big) \tag{72}$$

$$= \log\Big(1+\exp(\mu(\mathbf{x}))\Big) \tag{73}$$

$$q(y) = 1 \tag{74}$$

using (28) (75)

$$\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n}\left[\log\Big(\tfrac{[\nu(\mathbf{x}_i)]}{[1-\nu(\mathbf{x}_i)]}\Big)\cdot y_i - \log\Big([\tfrac{1}{1-\nu(\mathbf{x}_i)}]\Big)\right] \qquad \because \log 1 = 0 \tag{76}$$

$$= -\sum_{i=1}^{i=n}\left[\langle\mathbf{x}_i,\mathbf{w}\rangle\cdot y_i - \log\Big(1+\exp(\langle\mathbf{x}_i,\mathbf{w}\rangle)\Big)\right] \tag{77}$$

$$\nabla\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n}\left[\mathbf{x}_i\cdot y_i - \tfrac{\mathbf{x}_i\exp(\langle\mathbf{x}_i,\mathbf{w}\rangle)}{1+\exp(\langle\mathbf{x}_i,\mathbf{w}\rangle)}\right] \tag{78}$$

$$= -\sum_{i=1}^{i=n}\mathbf{x}_i\left[y_i - \tfrac{1}{1+\exp(-\langle\mathbf{x}_i,\mathbf{w}\rangle)}\right] \tag{79}$$

$$= -\sum_{i=1}^{i=n}\mathbf{x}_i\Big[y_i - \nu(\mathbf{x}_i)\Big] \tag{80}$$

5. (2 pts) Now let us tackle something new. Let

$$\Pr(\mathsf{Y}=y|\mathbf{x}) = \frac{[\nu(\mathbf{x})]^y}{y!}\exp(-\nu(\mathbf{x})), \quad \text{where} \quad y=0,1,2,\ldots. \tag{81}$$

Identify the functions $\mu(\mathbf{x})$, $\lambda(\mathbf{x})$ and $q(y)$ for the above specialization. Based on the linear parameterization in Ex 2.2, derive the negative log-likelihood and gradient. [Hint: Y here follows the Poisson distribution, which is useful for modeling integer-valued events, e.g., the number of customers at a given time.]

Ans: rewriting (82) in a form similar to (23),

$$\Pr(\mathsf{Y}=y|\mathbf{x}) = \frac{[\nu(\mathbf{x})]^y}{y!}\exp(-\nu(\mathbf{x})) \tag{82}$$
$$= \frac{\exp\Big(y\cdot\log(\nu(\mathbf{x}))\Big)}{y!}\exp(-\nu(\mathbf{x})) \tag{83}$$
$$= \tfrac{1}{y!}\exp\Big(y\cdot\log(\nu(\mathbf{x})) - \nu(\mathbf{x})\Big) \tag{84}$$

$$\mu(\mathbf{x}) = \log(\nu(\mathbf{x})) \tag{85}$$
$$\lambda(\mathbf{x}) = \nu(\mathbf{x}) \tag{86}$$
$$q(y) = \tfrac{1}{y!} \tag{87}$$
$$\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n}\Big[\mu(\mathbf{x}_i)\cdot y_i - \lambda(\mathbf{x}_i) + \log q(y_i)\Big] \tag{88}$$
$$= -\sum_{i=1}^{i=n}\Big[\langle\mathbf{x}_i,\mathbf{w}\rangle\cdot y_i - exp(\langle\mathbf{x}_i,\mathbf{w}\rangle) - \log(y_i!)\Big] \tag{89}$$
$$\nabla\ell_n(\mathbf{w}) = -\sum_{i=1}^{i=n}\Big[\mathbf{x}_i\cdot y_i - \mathbf{x}_i\cdot exp(\langle\mathbf{x}_i,\mathbf{w}\rangle)\Big] \tag{90}$$
$$= -\sum_{i=1}^{i=n}\Big[\mathbf{x}_i\cdot\Big(y_i - exp(\langle\mathbf{x}_i,\mathbf{w}\rangle)\Big)\Big] \tag{91}$$
$$= -\sum_{i=1}^{i=n}\Big[\mathbf{x}_i\cdot\Big(y_i - \nu(\mathbf{x}_i)\Big)\Big] \tag{92}$$
$$\tag{93}$$

Exercise 3: Ordinal regression (4 pts)

In many applications, the "labels" have an inherent order. For example, the letter grade A is preferred to B, which is preferred to C, etc. More generally, consider c ordinal labels $1, 2, \ldots, c$, where we prefer label k than $k+1$, for each $k = 1, \ldots, c-1$. [The preference is transitive, i.e., any "smaller" label is preferred over a "larger" label.]

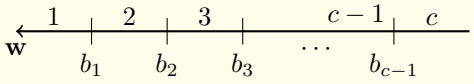
1. (2 pts) Let us consider $c-1$ *parallel* hyperplanes $H_k := \{\mathbf{x} : \langle\mathbf{x},\mathbf{w}\rangle + b_k = 0\}$, which partition our space into c rectangular regions. We define our prediction as

$$\hat{y} \le k \iff \langle\mathbf{x},\mathbf{w}\rangle + b_k > 0, \tag{94}$$

or more explicitly,

$$\hat{y} = k \iff [\langle\mathbf{x},\mathbf{w}\rangle + b_k > 0 \text{ and } \langle\mathbf{x},\mathbf{w}\rangle + b_{k-1} \le 0], \tag{95}$$

where $b_0 := -\infty$ and $b_c := \infty$.



The ordering in the labels is now respected, if we constrain $b_1 \le b_2 \le \cdots \le b_{c-1}$:

$$\hat{y} \le k \implies \hat{y} \le l, \quad \forall\, l \ge k. \tag{96}$$

We learn the weights $\mathbf{w}$ and $b_1, \dots, b_{c-1}$ by reducing to a sequence of (coupled) binary classifications:

$$\min_{\mathbf{w}, b_1 \le b_2 \le \cdots \le b_{c-1}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \max\{0, 1 - (\llbracket y_i = k \rrbracket - \llbracket y_i = k+1 \rrbracket)(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\}, \tag{97}$$

where recall that $\llbracket A \rrbracket$ is 1 if A is true and 0 otherwise. It is clear that when $c = 2$, the above reduces to the familiar soft-margin SVM. Derive the Lagrangian dual of (97). [If it helps, you may ignore the constraint $b_1 \le \ldots \le b_{c-1}$.]

Ans: Let $C_{ik} = \llbracket y_i = k \rrbracket - \llbracket y_i = k+1 \rrbracket$

The constraints $b_1 \le \ldots \le b_{c-1}$, can be written as $b_j - b_{j+1} \le 0, \forall j \in \{1, \dots, c-2\}$

(98)

Then the Lagrangian dual of equation-(97) is (99)

$$\min_{\mathbf{w}, b_1 \le b_2 \ldots \le b_{c-1}} \max_{0 \le \alpha_{ik} \le 1} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik}[1 - C_{ik}(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)] + \sum_{j=1}^{c-2} \gamma_j (b_j - b_{j+1}) \tag{100}$$

Swap min and max (101)

$$\max_{0 \le \alpha_{ik} \le 1} \min_{\mathbf{w}, b_1 \le b_2 \ldots \le b_{c-1}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik}[1 - C_{ik}(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)] + \sum_{j=1}^{c-2} \gamma_j (b_j - b_{j+1}) \tag{102}$$

(103)

We minimize the inner optimiziation by taking partial derivative with respect to inner arguments.

$$\frac{\partial}{\partial \mathbf{w}} = \lambda \mathbf{w} - \sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik} C_{ik} \mathbf{x}_i = 0 \tag{104}$$

$$\implies \mathbf{w} = \frac{1}{\lambda}\sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik} C_{ik} \mathbf{x}_i \tag{105}$$

Now , maybe : we can ignore the constraints $b_1 \le b_2 \ldots \le b_{c-2}$, The last summation term would be zero , hence the $\frac{\partial}{\partial b_k} =$
$\frac{\partial}{\partial b_k} = -\sum_{i=1}^{n} \alpha_{ik} C_{ik} = 0 \forall \in 1, 2, \ldots c-1$
We sub the value of $\mathbf{w}, b_i$ in

$$\max_{0 \le \alpha_{ik} \le 1} \min_{\mathbf{w}, b_1 \le b_2 \ldots \le b_{c-1}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik}[1 - C_{ik}(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)] + \sum_{j=1}^{c-2} \gamma_j (b_j - b_{j+1}) \tag{106}$$

Dual is then (107)

$$\max_{0 \le \alpha_{ik} \le 1} \frac{\lambda}{2}\|\frac{1}{\lambda}\sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik} C_{ik} \mathbf{x}_i\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik}[1 - C_{ik}(\langle \mathbf{x}_i, \frac{1}{\lambda}\sum_{k=1}^{c-1}\sum_{i=1}^{n} \alpha_{ik} C_{ik} \mathbf{x}_i\rangle + b_k)] + \sum_{j=1}^{c-2} \gamma_j (b_j - b_{j+1}) \tag{108}$$

2. (2 pts) In the previous formulation, to learn b_k, essentially we take class k as positive and class $k+1$ as negative. Can you find a "better" alternative? Write down the formulation. [Hint: it would be similar to (97).]

Ans: We have, $\sum_{k=1}^{c-1}\sum_{i=1}^{n} \max\{0, 1 - (\llbracket y_i = k \rrbracket - \llbracket y_i = k+1 \rrbracket)(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\}$
if , $y_i = k$ and predicted correctly , then

$\max\{0, 1 - ([\![y_i = k]\!] - [\![y_i = k+1]\!])(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\} = \max\{0, 1 - (\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\}$.
Penalty paid is zero if $\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k \geq 1$ or paid some penalty if $\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k < 1$.
Similarly for true label $y_i = k+1$,we have
$\max\{0, 1 - ([\![y_i = k]\!] - [\![y_i = k+1]\!])(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\} = \max\{0, 1 + (\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\}$
We pay penalty if we wrongly classify the label as k i.e $(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k) > 0$.
Issue : for a hyperplane $\langle \mathbf{x}, \mathbf{w}\rangle$, if the true label $y_i \neq k$ or $_i \neq k+1$ we have $([\![y_i = k]\!] = [\![y_i = k+1]\!]) = 0$ and therefore the penalty is always 1 regardless if it lies on correct side of hyperplane or not.We can modify this to

$$\min_{\mathbf{w}, b_1 \leq b_2 \leq \cdots \leq b_{c-1}} \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{k=1}^{c-1}\sum_{i=1}^{n} \max\{0, 1 - ([\![y_i \leq k]\!] - [\![y_i \geq k+1]\!])(\langle \mathbf{x}_i, \mathbf{w}\rangle + b_k)\}, \tag{109}$$

To take all the points in the left or right of all the labels into consideration not just the label to the left or right .