Internal Insensitive Loss for Ordinal Classification

Structure of Code:
The following steps include the basic structure of code --

1. Read the entire data from the file.
2. Calculate the total number of classes or bins based on the provided bin size and size of data.
   number_of_classes = data_size / bin_size
3. Define the ranges (left extreme and right extreme) for each bin.
4. Segregate the data into files as “train” and “test” using Kfold algorithm.
5. Calculate values of different variables in the dual form and provide it to QP solver to find the Lagrangian constants **.**
6. Find weight vector using Lagrangian constants.
7. Classify the input samples using the obtained weight vector and determine accuarcy and mean absolute error loss.
8. Plot the graph based on the above results.

There are two files namely, binning.py and computation.py
binning.py : steps 1 to 4
computation.py : steps 5 to 8

Running the code:
python computation.py bin_size
Example: python computation.py 50

Conversion of given primal to dual:

$$f = \frac{\lambda}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i + \xi_i^*$$

s.t.

$$\xi_i \geq y_\ell^i - y + \langle x^i, w\rangle (y - y_\ell^i) + b_y - b_{y_\ell^i}$$

$$\xi_i^* \geq y - y_r^i + \langle x^i, w\rangle (y - y_r^i) + b_y - b_{y_r^i}$$

$$\xi_i \geq 0 \qquad \xi_i^* \geq 0.$$

$\Rightarrow$

$$L = \frac{\lambda}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i + \xi_i^*$$
$$- \sum_{i=1}^{m}\alpha_i\left(\xi_i - [y_\ell^i - y + b_y - b_{y_\ell^i} + \langle x^i, w\rangle (y - y_\ell^i)]\right)$$
$$- \sum_{i=1}^{m}\alpha_i^*\left(\xi_i^* - [y - y_r^i + \langle x^i, w\rangle (y - y_r^i) + b_y - b_{y_r^i}]\right)$$
$$- \sum_{i=1}^{m}\delta_i\xi_i - \sum_{i=1}^{m}\delta_i^*\xi_i^*$$

$$\frac{\delta L}{\delta \xi_i} = \frac{C}{m} - \alpha_i - \delta_i = 0 \qquad \text{---- (1)}$$

$$\frac{\delta L}{\delta \xi_i^*} = \frac{C}{m} - \alpha_i^* - \delta_i^* = 0 \qquad \text{---- (2)}$$

$$\frac{\delta L}{\delta w} = \lambda w + \sum \alpha_i x^i (y - y_\ell^i) + \alpha_i^* x^i (y - y_u^i)$$

$$w = \frac{-1}{\lambda}\sum \alpha_i x^i (y - y_\ell^i) + \alpha_i^* x^i (y - y_u^i).$$

Let $A_i = \alpha_i (y - y_\ell^i) + \alpha^* (y - y_u^i)$.

$$L = \frac{\lambda}{2}\|w\|^2 + \sum \alpha_i \left[y_l^i - y + b_y - b_{y_l^i} + \langle x^i, w\rangle (y - y_l^i)\right]$$

$$+ \sum \alpha_i^* \left[y - y_r^i + b_y - b_{y_r^i} + \langle x^i, w\rangle (y - y_r^i)\right]$$

- [Let $B_i = \alpha_i (y_l^i - y) + \alpha_i^* (y - y_r^i)$]

$$L = \frac{\lambda}{2}\|w\|^2 + \sum B_i + \sum \alpha_i (b_y - b_{y_l^i}) + \alpha_i^* (b_y - b_{y_r^i})$$

$$+ \sum \langle x^i, w\rangle A_i$$

- from (3).

$$L = \frac{\lambda}{2}\|w\|^2 + \sum B_i - \sum \alpha_i b_{y_l^i} + \alpha_i^* b_{y_r^i}$$

$$+ \sum \langle x^i, w\rangle A_i$$

from (4) $\quad w = -\frac{1}{\lambda} \sum A_i x_i \qquad \|w\|^2 = w^T w$

$$L = \frac{\lambda}{2}\left(\frac{-1}{\lambda}\right)^2 \sum_i \sum_j (A_i x_i)^T (A_j x_j) + \sum B_i +$$

$$-\frac{\lambda}{\lambda} \sum_i \sum_j (A_j x_j)^T x_i A_i - \sum \alpha_i (b_y - b_{y_l^i}) - \sum \alpha_i^* (b_y - b_{y_r^i})$$

$$L = \frac{1}{2\lambda} \sum_i \sum_j x_i^T A_i^T A_j x_j - \frac{1}{\lambda} \sum_i \sum_j x_j^T A_j^T x_i A_i$$

$$- \sum_i (\alpha_i b_{y_l^i} + \alpha_i^* b_{y_r^i}) + \sum_i B_i$$

$$= \sum_i B_i - \sum_i \alpha_i b_{y_l^i} - \sum_i \alpha_i^* b_{y_r^i} - \frac{1}{2\lambda} \sum_i \sum_j x_i^T A_i^T A_j x_j$$

$$L = \sum_i \alpha_i (y_l^i - y) + \alpha_i^* (y - y_r^i) - \sum_i \alpha_i b_{y_l^i} - \sum_i \alpha_i^* b_{y_r^i}$$

$$\frac{-1}{2\lambda} \sum_i \sum_j x_i^T C' x_j$$

where $C' = A_i^T A_j = \left[\alpha_i (y - y_l^i) + \alpha_i^* (y - y_r^i)\right]^T [\quad]$

$$C' = A_i A_j$$

$$= \left(\left\{ \alpha_i (y - y_l^i) + \alpha_i^* (y - y_r^i) \right\} \left\{ \alpha_j (y - y_l^j) + \alpha_j^* (y - y_r^j) \right\} \right)$$

$$x_i^T C x_j = x_i^T \alpha_i \alpha_j (y - y_l^i)(y - y_l^j) x_j +$$

$$x_i^T \alpha_i \alpha_j^* (y - y_l^i)(y - y_r^j) x_j +$$

$$x_i^T \alpha_i^* \alpha_j (y - y_r^i)(y - y_l^j) x_j +$$

$$x_i^T \alpha_i^* \alpha_j^* (y - y_r^i)(y - y_r^j) x_j$$

$$\frac{1}{2}\alpha^T Q \alpha + P^T \alpha \qquad A\alpha = b \qquad G\alpha \le h.$$

$$\alpha = \begin{bmatrix} \alpha_1 \alpha_2 \cdots \alpha_m \alpha_1^* \alpha_2^* \cdots \alpha_m^* \end{bmatrix}_{1 \times 2m}$$

$$P = \begin{bmatrix} y_1^1 - y - by_1^1 \; y_1^2 - y - by_1^2 \cdots y_1^n - y - by_1^n \; y_2^1 - y - by_2^1 \cdots y_2^n - y - by_2^n \end{bmatrix}_{1 \times 2m}$$

$$Q = \begin{bmatrix} ① & ② \\ ③ & ④ \end{bmatrix}$$

① $Q_{ij} = \sum_i \sum_j x_i^T x_j \left(y - y_1^i\right)\left(y - y_1^j\right) \quad \forall i = 1, 2 \ldots m \quad j = 1, 2 \ldots m$

② $Q_{ij} = \sum_i \sum_j x_i^T x_j \left(y - y_2^i\right)\left(y - y_2^j\right) \quad \forall i = m+1, \ldots 2m \quad j = m+1, \ldots 2m$

③ $Q_{ij} = \sum_i \sum_j x_i^T x_j \left(y - y_1^i\right)\left(y - y_2^j\right) \quad \forall i = 1, \ldots m \quad j = m+1, \ldots 2m$

④ $Q_{ij} = \sum_i \sum_j x_i^T x_j \left(y - y_2^i\right)\left(y - y_1^j\right) \quad \forall i = m+1, \ldots 2m \quad j = 1, \ldots m$

$$Ax = b \qquad Gx \le h$$

$$\sum_i \alpha_i + \alpha_i^* = 0 \qquad \alpha_i, \alpha_i^* \le \frac{C}{m}$$

$$A = \begin{bmatrix} 1 \ldots 1 \end{bmatrix}_{1 \times 2m} \qquad G = I_{2m \times 2m}$$

$$b = \begin{bmatrix} 0 \end{bmatrix}_{1 \times 1} \qquad h = \begin{bmatrix} C/m \\ \vdots \\ C/m \end{bmatrix}_{2m \times 1}$$

List of files written on our own:
binning.py and computation.py