Pattern Recognition: Homework 3

Due date: 2023.3.14

Problem 1 (20 pt)

We have learned three different kinds of linear methods: LDA(Fisher), logistic regression and SVM. There are many similarities between them, i.e. all the methods share the same parameter $\boldsymbol{w}, b$. But there are also some subtle differences. We can view them in a unified general framework: they are essentially the same algorithm minimizing following loss

$$\mathcal{L}(\boldsymbol{w}, b) = \frac{1}{N} \sum_{i=1}^{N} \ell(\boldsymbol{w}^\top \boldsymbol{x}_i + b, y_i).$$

However, LDA (using Fisher's condition), logistic regression and SVM use different single-data loss function $\ell(\hat{y}, y) : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^+$. Please find out the concrete form of loss function (like $\ell(\cdot, y = 1)$) for each algorithm and plot its curve. How does that affect these methods' behavior and make them different?

Problem 2 (20 pt)

In our class, we have learned the primal form of SVM problem and its dual form. Please prove that, solving the **hard-margin** problem

$$\begin{aligned} &\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ s.t. \quad &y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, 1 \leq i \leq n. \end{aligned}$$

is equivalent to

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ s.t. \quad &\sum_{i=1}^{n} \alpha_i y_i = 0, \\ &\alpha_i \geq 0, \quad 1 \leq i \leq n. \end{aligned}$$

Namely, denote the solution in the first problem as $\boldsymbol{w}^*$ and solution in the second problem as $\boldsymbol{\alpha}^* = (\alpha_1^*, \ldots, \alpha_n^*)$. Prove that the solutions satisfy

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i,$$

and

$$\alpha_i^* > 0 \Longleftrightarrow \boldsymbol{w}^{*\top} \boldsymbol{x}_i + b = y_i.$$

hint: Consider the Lagrangian.

Problem 3 (10 pt)

Search the paper and conduct a literature review for how to add different regularizations to SVM, and more sophisticated forms of kernel function.

Problem 4 (50 pt)

In this section, you will write your own SVM to finish a very simple face recognition problem. In the folder `face_data` there are two sub-folders `0` and `1`, each containing 300 faces of an individual person. Randomly separate 250 vs 50 as the training set and test set for each person's face images. You may need package like `opencv-python` or `Pillow` for loading these images as a set of high-dimensional vectors. Please search for their documents for usage.

- Write SVM to classify them on the training set, test and report the accuracy on the test set.
- Try different kernel functions and penalty strength C, and report their difference in accuracy. Which composition is the best?
- Which are the supporting vectors (images) in the training set? (find those point that are classified correctly with non-zero α_i). Plot them are explain the potential reason why they become the supporting vectors. (open problem)

Good News: You can use package like `scipy.svm` for this homework. Writing your own SVM can get 20 points bonus but is not required. If you do so, please point it out in your report proudly.