Fall 2022 CS165B: Introduction to Machine Learning – Homework 5

Due: **Tuesday, Dec 6th, 11:59 pm PST**

Note: Please upload your written part as a PDF, and please upload your code as a separate file in (".py" or ".ipynb") form. Both files should be uploaded to Gauchospace.

1. (30 points) In lecture, we discussed that the linear SVM problem can be solved by primal QP algorithm. Show that the matrix Q described in the linear SVM algorithm is positive semi-definite (that is $\mathbf{u}^T Q \mathbf{u} \geq 0$ for any $\mathbf{u}$).

   The result means that the QP-problem is convex. Convexity is useful because this makes it easy to find an optimal solution.

2. (35 points) Consider a dataset with three data points in $\mathbb{R}^2$:

$$X = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ -2 & 0 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix}$$

   Manually solve the SVM to get the optimal hyperplane $(b^*, \mathbf{w}^*)$ and its margin.

   The objective function is given below:

$$\begin{aligned} \text{minimize}_{\{b,\mathbf{w}\}} : \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to:} \quad & y_n\left(\mathbf{w}^T\mathbf{x}_n + b\right) \geq 1 \quad (n = 1, \cdots, N) \end{aligned}$$

3. (35 points) Experiment: Implement hard margin SVM using Python.

   (a) Use Python to implement hard margin SVM. Use the CVXOPT (`http://cvxopt.org`) as the QP solver.

   Note: The notations used in the CVXOPT tutorial (`http://cvxopt.org/userguide/coneprog.html#quadratic-programming`) is different from what we learned in class. Specifically, we used (Q, $\mathbf{p}$, A, $\mathbf{c}$) in class and the corresponding notations in CVXOPT are (P, $\mathbf{q}$, G, $\mathbf{h}$). Verify your results using the toy data in class.

   (b) Use random datasets (X, y) as input to your algorithm and vary the sample size and feature dimensionality. Investigate how time costs grow as you increase the sample size (while fix the dimensionality), and as you increase the dimensionality (while fix the sample size). Is your algorithm efficient?