CSE 404: Introduction to Machine Learning (Fall 2019)
Homework #9
Due 11/22/2019 11:59p

Note: (1) LFD refers to the textbook "Learning from Data".

1. (20 points) Exercise 8.5 (e-Chap:8-11) in LFD.

   Show that the matrix Q described in the linear hard-margin SVM algorithm above is positive semi-definite (that is $\boldsymbol{u}^T Q \boldsymbol{u} \geq 0$ for any $\boldsymbol{u}$).

   Note: The result means that the QP-problem is convex. Convexity is useful because this makes it easy to find an optimal solution. In fact, standard QP-solvers can solve our convex QP-problem in $O((N+d)^3)$.

2. (20 points) Exercise 8.11 (e-Chap:8-28) in LFD.

   (a) Show that the problem in (8.21) is a standard QP-problem:

   $$\min_{\boldsymbol{\alpha} \in \mathbf{R}^N} \frac{1}{2} \boldsymbol{\alpha}^T Q_D \boldsymbol{\alpha} - \mathbf{1}_N^T \boldsymbol{\alpha}$$
   $$\text{subject to: } A_D \boldsymbol{\alpha} \geq \mathbf{0}_{N+2}$$

   where Q_D and A_D (D for dual) are given by

   $$Q_D = \begin{bmatrix} y_1 y_1 \mathbf{x}_1^T \mathbf{x}_1 & \dots & y_1 y_N \mathbf{x}_1^T \mathbf{x}_N \\ y_2 y_1 \mathbf{x}_2^T \mathbf{x}_1 & \dots & y_2 y_N \mathbf{x}_2^T \mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_N y_1 \mathbf{x}_N^T \mathbf{x}_1 & \dots & y_N y_N \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix} \text{ and } A_D = \begin{bmatrix} \mathbf{y}^T \\ -\mathbf{y}^T \\ I_{N \times N} \end{bmatrix}$$

   [Hint: Recall that an equality corresponds to two inequalities.]

   (b) The matrix Q_D of quadratic coefficients is $[Q_d]_{mn} = y_m y_n \mathbf{x}_m^T \mathbf{x}_n$. Show that $Q_D = X_s X_s^T$, where X_s is the 'signed data matrix':

   $$X_s = \begin{bmatrix} -y_1 \mathbf{x}_1^T - \\ -y_2 \mathbf{x}_2^T - \\ \vdots \\ -y_N \mathbf{x}_N^T - \end{bmatrix}$$

   Hence, show that Q_D is positive semi-definite. This implies that the QP-problem is convex.

3. (20 points) Problem 8.1 (e-Chap:8-46) in LFD.

   Consider a data set with two data points $\mathbf{x}_{\pm} \in \mathbb{R}^d$ having class ± 1 respectively. Manually solve (8.4) by explicitly minimizing $\|w\|^2$ subject to the two separation constraints.

   Compute the optimal (maximum margin) hyperplane $(b^*, \mathbf{w}^*)$ and its margin. Compare with your solution to Exercise 8.1.

4. (20 points) Problem 8.2 (e-Chap:8-46) in LFD.

   Consider a data set 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 (8.4) to get the optimal hyperplane $(b^*, \mathbf{w}^*)$ and its margin.

5. (20 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?