CS 412: Introduction to Machine Learning

Homework 4

Prof.: Sathya N. Ravi Assigned: 04/02/2022, Due: 04/21/2022

Few reminders:

- Homework is due at the end of day on the designated date on Blackboard.
- No homework or project is accepted in mailbox of instructor.
- You may discuss homework with classmates and work in groups of up to **two**. However, you may not share any code, carry out the assignment together, or copy solutions from any other groups. Discussions between groups should be minimal, verbal during lecture or if appropriate, on Campuswire only. The submitted version must be worked out, written, and submitted by your group alone.
- **Important:** Each question (or subpart) should have a **lead** who has finalized the submitted solutions after discussions within group. The lead should be *explicitly* indicated in the submissions. There can be two leads for any question. The submitted solution is assumed to be verified by the other person, and so the grade is assigned equally.
- All solutions must be typeset (I recommend Latex, Markdown, or Word, please do not use nonstandard formats) with code attached in Python format whichever is appropriate. Code in Jupyter Notebook converted to Pdf for Gradescope and more experimental results including figures should be provided in the report.
- Your final submission will be made in two places: (i) zip folder with a PDF file containing solutions for each question including text, sample figures, calculation, analysis, and general writeup submitted to Blackboard; and (ii) one PDF submitted to Gradescope. **Everyone** will make these two submissions.
- Submitting someone else's work (outside of the group) as your own is academic misconduct. Such cheating and plagiarism will be dealt with in accordance with University procedures (see the page on Academic Misconduct at this link).

Answer the following questions as completely as possible. We use $\mathbb{R}^d$ to denote d dimensional vector spaces over the reals $\mathbb{R}$. Similarly, $\mathbb{Z}_+ =$ Set of positive integers, $\mathbb{Z}_{\geq 0} =$ set of nonnegative integers.

1. **Dual Soft Margin SVM Formulation.** Please go through lectures 19, 20, and 21 before you start working on this problem. In this problem, we will derive the dual formulation of the Soft Margin SVM (SM-SVM from now on) step by step. Recall the setup: you are given with n datapoints $x_i, y_i, i = 1, ..., n$ where $x_i \in \mathbb{R}^d, y_i \in \{+1, -1\}$. SM-SVM is defined using a hyperplane $w \in \mathbb{R}^d$, constant shift $b \in \mathbb{R}$, nonnegative slack variables $s \in \mathbb{R}^n_{\geq 0}$. The optimal w^*, b^*, s^* are unknown parameters that can be found by solving the following constrained optimization problem:

$$\min_{w,b,s} \frac{1}{2}\|w\|_2^2 \quad \text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - s_i, s_i \geq 0 \; \forall \; i = 1, ..., n. \tag{1}$$

   (a) Introduce two dual variables $\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}^n$ for the two types of constraints in (1). That is, we associate α_i with the constraint $y_i(w^T x_i + b) \geq 1 - s_i$, and similarly β_i with the constraint $s_i \geq 0$. Write down the Lagrangian function $L(w, b, s, \alpha, \beta)$. Please make sure you have the correct sign and constraint for the dual variables.

   (b) Similar to the Hard Margin case, we have to differentiate the lagrangian with respect to each of the primal variables w, b, s. Write down the partial derivatives $\frac{\partial L}{\partial w}, \frac{\partial L}{\partial b}, \frac{\partial L}{\partial s}$ of L with respect to each of w, b, s respectively.

   (c) Set the partial derivatives computed in part (b) to zero, and express the primal variables in terms of dual variables by rearranging. Eliminate the primal variables w, b, s, and write down the dual SM-SVM problem. Remember not to forget the constraint on the dual variables. Hint: Upon doing the above three steps correctly, your dual SM-SVM formulation should coincide with the one given in Slide 11 of Lecture 21.

2. **Solving SVMs On Paper.** Consider the dataset containing 8 data points given by: $(2, 9, +1)$, $(4, 6, +1), (4, 10, +1), (-4, 0, +1), (10, 0, -1), (14, 3.5, -1), (\frac{13}{3}, 0, -1), (11, 5, -1)$ where the last coordinate represents the label, so there are four points each in positive, and negative classes respectively.

   (a) Draw the data points above, and sketch the maximum margin hyperplane and also the marginal hyperplane (the hyperplane parallel to the maximum margin hyperplane which passes through the nearest points to the maximum margin hyperplane). Write down the value of the margin (i.e., the distance from the decision boundary to the margin boundary).

   (b) Choose three support vectors, write out the system of equations for those support vector datapoints. That is, these points satisfy the inequality constraints as a equality, and so, for these points, $y_i(w_1 x_{i,1} + w_2 x_{i,2} + b) = 1$. Solve for w_1, w_2, b.

   (c) Will the optimal w^* change if we remove: (i) one point $(\frac{13}{3}, 0, -1)$ from the dataset?, (ii) two points $(\frac{13}{3}, 0, -1), (11, 5, -1)$ from the dataset?

3. **Comparing Logistic Regression and SVM by Iterations.** Please download, and preprocess the Spambase dataset from `https://archive.ics.uci.edu/ml/datasets/spambase`. As usual, we will use $x_i \in \mathbb{R}^d$ to represent individual data points in $d-$dimensions, and $y_i \in \{+1, -1\}$. It is convenient to "assemble" them in to a data matrix $\hat{X} \in \mathbb{R}^{n \times d+1}$ where d is the dimension of individual data points with the last column containing the label vector $y \in \mathbb{R}^n$, and n is the total number of individual data points in the dataset. We will use $X \in \mathbb{R}^{n \times d}$ to denote the first d columns of $\hat{X}$ for simplicity.

   To use gradient descent type algorithms, we have to write the dual SM-SVM formulation in a form for which we can compute gradients. Whence, we will use the following unconstrained formulation given by the so-called "hinge" loss as,

$$\min_{\alpha} \frac{1}{2} \alpha^T K \alpha + \frac{\lambda}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i (K\alpha)_i\right), \tag{2}$$

   where $K \in \mathbb{R}^{n \times n}$ represents the Kernel matrix (so $K\alpha \in \mathbb{R}^n$ is a vector), and $(K\alpha)_i$ represents the $i-$th coordinate of the vector $K\alpha \in \mathbb{R}^n$. The goal of the problem is to solve the Hinge Loss SVM problem (2) using iterative algorithms such as gradient descent. We will proceed step-by-step.

   (a) Given specific values of the matrix K, vector α, scalar λ, fill in the `hinge_loss` function that returns the objective function or loss value of problem (2).

   (b) Recall the definition gradient of $\max(x, 0)$ is given by $\nabla_x \max(x, 0) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & x \leq 0. \end{cases}$ Now, given specific values of K, α, λ, fill in the `hinge_gradient` function that returns the gradient of the objective function or loss value of problem (2) with respect to α.
   Hint: Use Chain rule. Note that $(K\alpha)_i$ can be written as $e_i^T (K\alpha) = e_i^T K \alpha$ where $e_i \in \mathbb{R}^n$ is the standard basis vector in $n-$dimensions i.e., the vector $e_i \in \mathbb{R}^n$ has a 1 in the $i-$th coordinate, and 0 elsewhere. What is gradient of $e_i^T K \alpha$ with respect to α?

   (c) **Forming K from X.** We will try various kernel functions. Recall that the $(i, j)-$th entry of the matrix K is given by $\phi(x_i)^T \phi(x_j)$ where $\phi : \mathbb{R}^d \to \mathbb{R}^D$ is the feature map that takes data points $x_i \in \mathbb{R}^d$ to higher dimensional feature vector $\phi(x_i) \in \mathbb{R}^D$. Given the features of the dataset $X \in \mathbb{R}^{n \times d}$, fill in the `linear_kernel` that implements Linear kernel, that is, $K^L_{ij} = x_i^T x_j$.
   Given X, the variance parameter $\gamma > 0$, fill in the `gaussian_kernel` that implements Gaussian kernel, that is, $K^G_{ij} = \exp(-\gamma \|x_i - x_j\|_2^2)$. Note that both functions will output the kernel matrix $K \in \mathbb{R}^{n \times n}$.

(d) Fix total number of iterations $T = 1000$. Fill in the code for training, plotting, and testing the model using gradient descent. Finally, print the test accuracy of the model. For plotting, the $x-$ axis corresponds to iterations t, and $y-$ axis corresponds to the loss value. You will also have to pick the step size or learning rate η, use $\eta = 0.1$, and the variance parameter γ of the Gaussian kernel K^G, use $\gamma = 1$.

(e) Similar to Homework 3, please report the accuracies of combinations of the two kernels, $T \in \{100, 200, 500, 1000\}, \eta \in \{0.01, 0.1, 0.2, 0.5\}$ for both Logistic Regression, and SVM. Which of the two is better? Provide all necesary experimental results to back up your claim. Explain the performance of the two algorithms (SVMs vs Logistic Regression) when using the two kernels.

Important Note. Follow the same instructions as Homework 1 and submit on Gradescope.
