Engineering Applications of Machine Learning and Data Analytics
Homework #3 [Due: 03/08/2021]

I acknowledge that this exam is solely my effort. I have done this work by myself. I have not consulted with others about this exam in any way. I have not received outside aid (outside of my own brain) on this exam. I understand that violation of these rules contradicts the class policy on academic integrity.

Name: ____________________________

Signature: ____________________________

Date: ____________________________

Instructions: There are four problems. X Partial credit is given for answers that are partially correct. No credit is given for answers that are wrong or illegible. Write neatly.

You must submit two PDFs on D2L. The first PDF has the results to the analytical questions as well as figures that are generated

Problem 1: __________ Problem 2: __________

Problem 3: __________ :

Total: __________

1 The ℓ_2 Support Vector Machine [20pts]

In class, we discussed that if our data is not linearly separable, then we need to modify our optimization problem to include slack variables. The formulation that was used is known as the ℓ_1-norm soft margin SVM. Now consider the formulation of the ℓ_2-norm soft margin SVM, which squares the slack variables within the sum. Notice that non-negativity of the slack variables has been removed.

$$\begin{aligned}
&\arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \\
&\qquad \text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \qquad \forall i \in [n]
\end{aligned}$$

Derive the dual form expression along with any constraints. Work must be shown. *Hints*: Refer to the methodology that was used in class to derive the dual form. The solution is given by:

$$\begin{aligned}
&\arg\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j - \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2 \\
&\qquad \text{s.t. } \alpha_i \geq 0 \quad \forall i \in [n] \quad \text{and} \quad \sum_{i=1}^{n}\alpha_i y_i = 0
\end{aligned}$$

$L = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum \xi_i^2 +$

2 Domain Adaptation Support Vector Machines [20pts]

We now look at a different type of SVM that is designed for domain adaptation and optimizes the hyperplanes given by $\mathbf{w}_S$ (source hyperplane) before optimizing $\mathbf{w}_T$ (target hyperplane). The process begins by training a support vector machine on source data then once data from the target are available, train a new SVM using the hyperplane from the first SVM and the data from the target to solve for a new "domain adaptation" SVM.

The primal optimization problem is given by

$$\begin{aligned} \arg\min_{\mathbf{w}_T,\xi} \quad & \frac{1}{2}\|\mathbf{w}_T\|^2 + C\sum_{i=1}^{n}\xi_i - B\mathbf{w}_T^T\mathbf{w}_S & \\ \text{s.t.} \quad & y_i(\mathbf{w}_T^T\mathbf{x}_i + b) \geq 1 - \xi_i & \forall i \in \{1,\dots,n\} \\ & \xi_i \geq 0 & \forall i \in \{1,\dots,n\} \end{aligned}$$

where $\mathbf{w}_S$ is hyperplane trained on the source data (*assumed to be known*), $\mathbf{w}_T$ is hyperplane for the target, $y_i \in \{\pm 1\}$ is the label for instance $\mathbf{x}_i$, C & B are regularization parameters defined by the user and ξ_i is a slack variable for instance $\mathbf{x}_i$. The problem becomes finding a hyperplane, $\mathbf{w}_T$, that minimizes the above objective function subject to the constraints. Solve/derive the dual optimization problem.

Note: I will give the class the solution to this problem prior to the due date because Problem #3 requires that you implement this algorithm in code.

3 Domain Adaptation SVM (Code) [20pts]

Implement the domain adaptation SVM from Problem #2. A data setfor the source and target domains (both training and testing) have been uploaded to D2L. There are several ways to implement this algorithm. If I were doing this for an assignment, I would implement the SVM (both the domain adaptation SVM and normal SVM) directly using quadratic programming. You do not need to build the classifier (i.e., solve for the bias term); however, you will need to find $\mathbf{w}_T$ and $\mathbf{w}_S$. To find the weight vectors, you will need to solve a quadratic programming problem and look through the documentation to learn how to solve this optiization task. The following Python packages are recommended:

- CVXOPT(https://cvxopt.org/)
- PyCVX (https://www.cvxpy.org/install/)

Note: Your solution can (and should) use any of the packages above.