Hull-Robert-HW3-Initial.md

- Robert Quinn Hull

- HW 3, ML
- 03/08/21

Outline

1. The l2 Support Vector Machines

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.

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

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:

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

Constraints

0.
$$\varepsilon_i \geq 0, -\varepsilon_i \leq 0$$
$$0 \leq \alpha_i \leq C$$

1.
$$y_i(\omega^T x_i + b) \geq 1 - \varepsilon_i$$

2.
$$y_i(\omega^T x_i + b) - 1 + \varepsilon_i \geq 0$$

Lagrangian

3.
$$L(\omega, b, \alpha, \varepsilon) = \frac{1}{2} \parallel \omega \parallel_2^2 + \frac{C}{2}\Sigma_{i=1}^n[\varepsilon_i^2] - \Sigma_{i=1}^n \alpha_i[y_i(\omega^T x_i + b) - 1 + \varepsilon_i]$$

Derivative

4.
$$\frac{dL}{d\omega} = w + 0 - \Sigma_{i=1}^n \alpha_i y_i x_i = 0 \rightarrow \omega = \Sigma_{i=1}^n \alpha_i y_i x_i$$

5.
$$\frac{dL}{db} = 0 + 0 - \Sigma_{i=1}^n \alpha_i y_i = 0 \rightarrow 0 = \Sigma_{i=1}^n \alpha_i y_i$$

6.
$$\frac{dL}{d\varepsilon} = 0 + C\Sigma_{i=1}^n \varepsilon_i - \Sigma_{i=1}^n \alpha_i = 0 \rightarrow \frac{1}{C}\Sigma_{i=1}^n \alpha_i = \Sigma_{i=1}^n \varepsilon_i$$

Back to Lagrangian

7.
$$L = \frac{1}{2}\Sigma_{i=1}^n \Sigma_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + \frac{C}{2}\Sigma_{i=1}^n \varepsilon_i^2$$
$$-\Sigma_{i=1}^n \alpha_i y_i \omega^T x_i - b\Sigma_{i=1}^n \alpha_i y_i + \Sigma_{i=1}^n \alpha_i - \Sigma_{i=1}^n \varepsilon_i$$

8.
$$L = \frac{1}{2}\Sigma_{i=1}^n \Sigma_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + \frac{C}{2}(\frac{1}{C})^2 \Sigma_{i=1}^n \alpha_i^2$$

$$-\Sigma_{i=1}^n \Sigma_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + \Sigma_{i=1}^n \alpha_i - \frac{1}{C}\Sigma_{i=1}^n \alpha_i$$

9.
$$L = (1 - \frac{1}{C})\Sigma_{i=1}^n \alpha_i - \frac{1}{2}\Sigma_{i=1}^n \Sigma_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + (\frac{1}{2C})\Sigma_{i=1}^n \alpha_i^2$$

Solution

10.
$$argmax_\alpha (1 - \frac{1}{C})\Sigma_{i=1}^n \alpha_i - \frac{1}{2}\Sigma_{i=1}^n \Sigma_{j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j + (\frac{1}{2C})\Sigma_{i=1}^n \alpha_i^2$$

s.t.
$$0 \leq \alpha_i \leq C$$
$$\Sigma_{i=1}^n \alpha_i y_i = 0$$

2. Domain Adaptation Support Vector Machines

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.

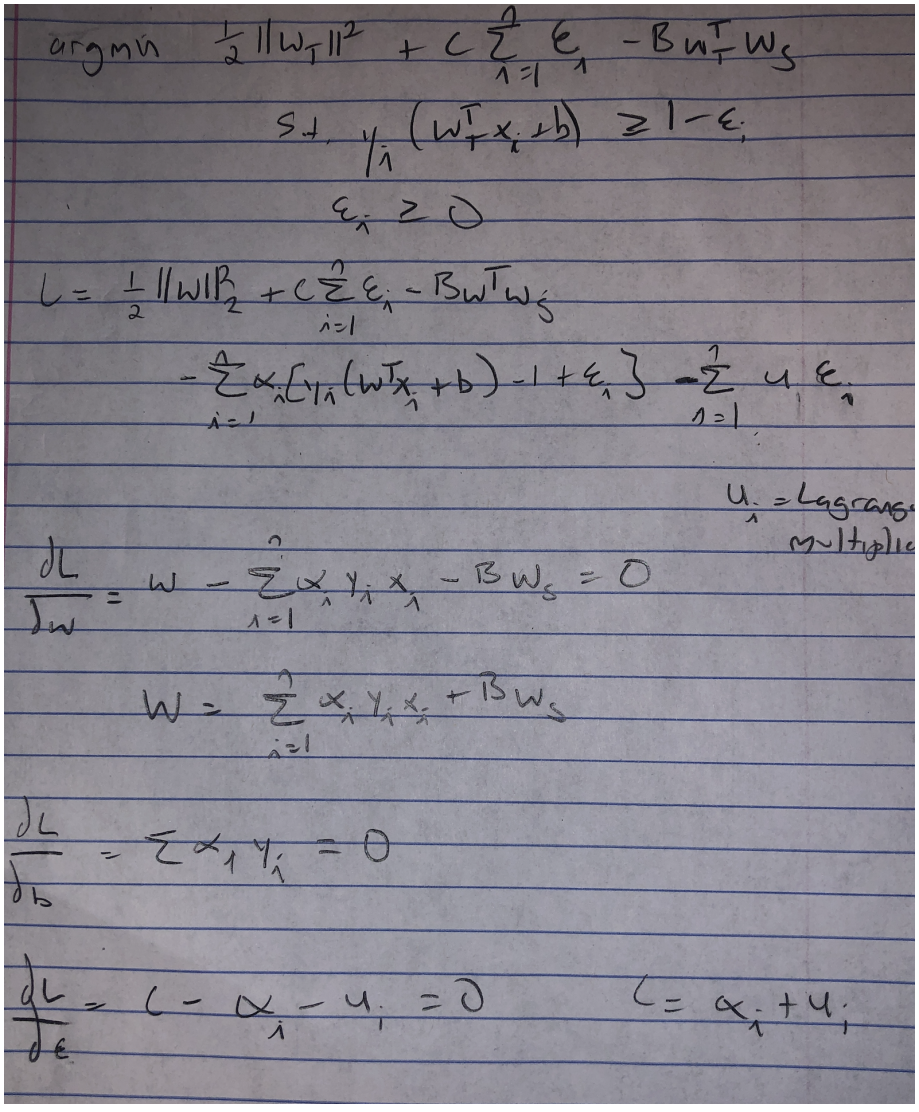
The primal optimization problem is given by

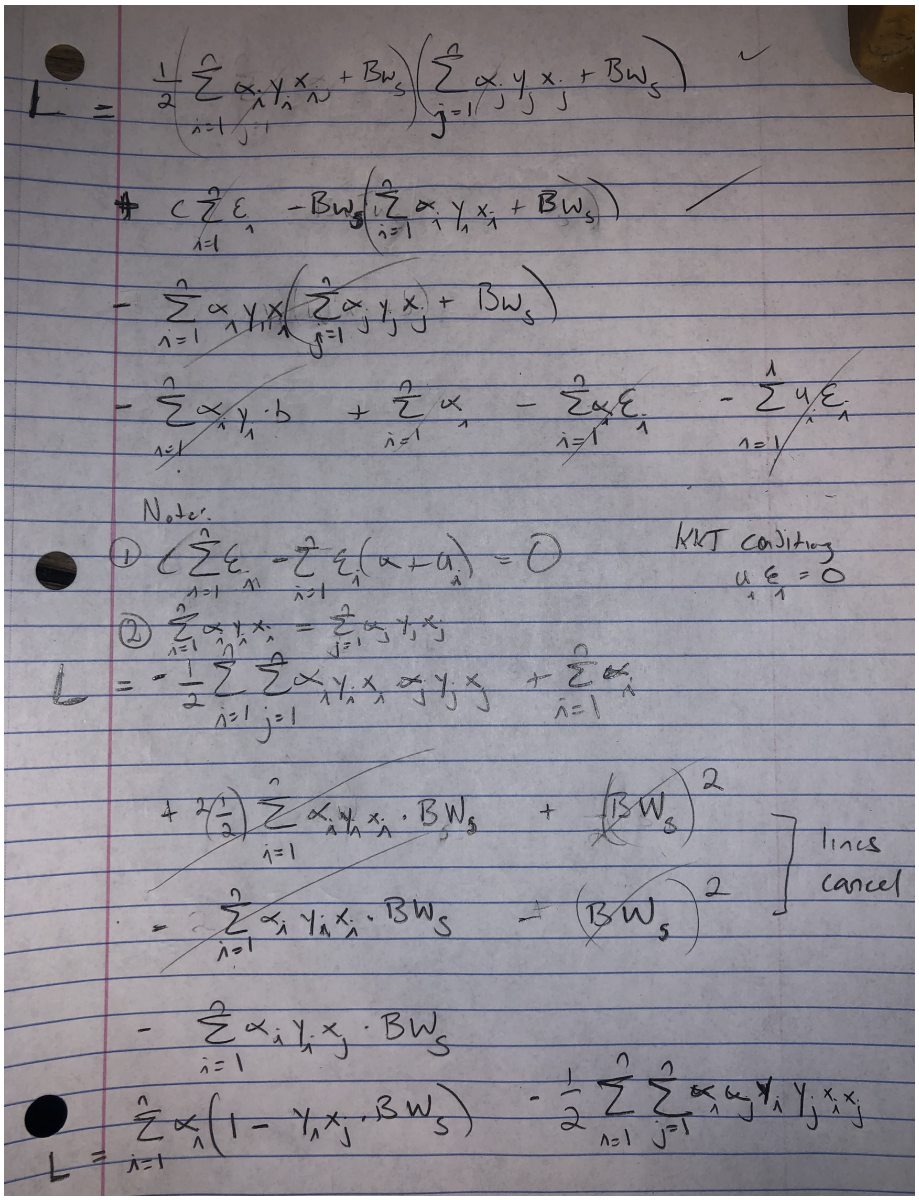
$$\begin{aligned} \arg\min_{\mathbf{w}_T,\xi} \quad & \frac{1}{2}\|\mathbf{w}_T\|^2 + C\sum_{i=1}^{n}\xi_i - B\mathbf{w}_T^T\mathbf{w}_S \\ \text{s.t.} \quad & y_i(\mathbf{w}_T^T\mathbf{x}_i + b) \geq 1 - \xi_i \qquad \forall i \in \{1,\ldots,n\} \\ & \xi_i \geq 0 \qquad \forall i \in \{1,\ldots,n\}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

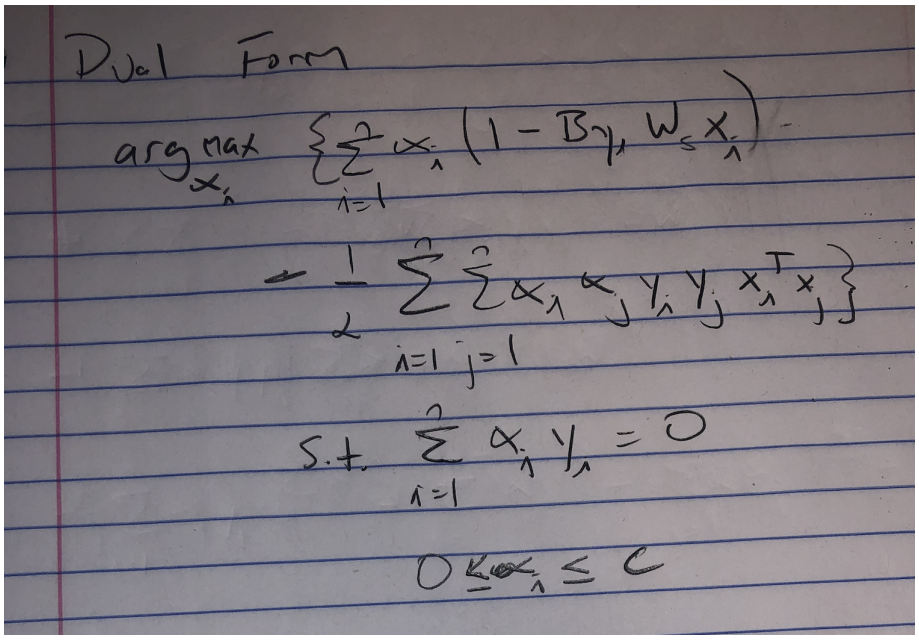
where $\mathbf{w}_S$ is hyperplane trained on the source data (*assumed to be known*), $\mathbf{w}_T$ is hyperplane for the target, $y_i \in \{\pm 1\}$ is the label for instance $\mathbf{x}_i$, C & B are regularization parameters defined by the user and ξ_i is a slack variable for instance $\mathbf{x}_i$. 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.

- sorry no latex : (







3. Density Estimation (Code)

3 Density Estimation (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 use any of the packages above.