COM S 5730 Homework 2

1. Please put required code files and report into a compressed file "HW2_FirstName_LastName.zip"
2. Unlimited number of submissions are allowed on Canvas and the latest one will be graded.
3. Due: **Tuesday Oct. 01, 2024 at 11:59pm.**
4. No later submission is accepted.
5. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
6. All students are required to typeset their reports using latex. Overleaf (`https://www.overleaf.com/learn/latex/Tutorials`) can be a good start.

---

1. (20 points) Consider the toy data set $\{([0,0],-1),([2,2],-1),([2,0],+1)\}$. Set up the dual problem for the toy data set. Then, solve the dual problem and compute α^*, the optimal Lagrange multipliers. (Note that there will be three weights $\boldsymbol{w} = [w_0, w_1, w_2]$ by considering the bias.)

Answer

Dual Problem Formulation

$$\max(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j)) \quad s.t. \quad \begin{cases} \sum_{j=1}^{n}\alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \end{cases}$$

Calculate the Kernel Matrix

Linear Kernel: $K(x_i, x_j) = x_i \cdot x_j$

$$K = \begin{bmatrix} x_1 \cdot x_1 & x_1 \cdot x_2 & x_1 \cdot x_3 \\ x_2 \cdot x_1 & x_2 \cdot x_2 & x_2 \cdot x_3 \\ x_3 \cdot x_1 & x_3 \cdot x_1 & x_3 \cdot x_3 \end{bmatrix}$$

Plug In the Values

$$\begin{aligned} x_1 &= (0,0) \quad y_1 = -1 \\ x_2 &= (2,2) \quad y_2 = -1 \\ x_3 &= (2,0) \quad y_3 = 1 \end{aligned}$$

$$\begin{aligned} x_1 \cdot x_1 &= (0\cdot 0) + (0 \cdot 0) = 0 \\ x_1 \cdot x_2 &= (0\cdot 2) + (0 \cdot 2) = 0 \\ x_1 \cdot x_3 &= (0\cdot 2) + (0 \cdot 0) = 0 \\ x_2 \cdot x_2 &= (2\cdot 2) + (2 \cdot 2) = 8 \\ x_2 \cdot x_3 &= (2\cdot 2) + (2 \cdot 0) = 4 \\ x_3 \cdot x_3 &= (2\cdot 2) + (0 \cdot 0) = 4 \end{aligned} \quad -> \quad K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 8 & 4 \\ 0 & 4 & 4 \end{bmatrix}$$

So, we have:

$$\max W(\alpha) = \sum_{i=1}^{3} \alpha_i - \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$= (\alpha_1 + \alpha_2 + \alpha_3) - \frac{1}{2}[0 + 0 + 0 + 0 + 8\alpha_2^2 - 4\alpha_2\alpha_3 - 4\alpha_2\alpha_3 + 4\alpha_3^2]$$

$$W(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 - 4\alpha_2^2 + 4\alpha_2\alpha_3 - 2\alpha_3^2$$

Constraints

(a) **Equality Constraint:** $\sum_{i=1}^{n} \alpha_i y_i = 0$

$$-\alpha_1 - \alpha_2 + \alpha_3 = 0$$

(b) **Non-Negative and Upper Bound Constraint:** $0 \leq \alpha_i \leq C \quad s.t. \quad C = 1$

Express α_1 in terms of α_2 and α_3:

$$\alpha_1 = -\alpha_2 + \alpha_3$$

Substitute α_1 in the Dual Function:

$$W(\alpha_2, \alpha_3) = (-\alpha_2 + \alpha_3) + \alpha_2 + \alpha_3 - 4\alpha_2^2 + 4\alpha_2\alpha_3 - 2\alpha_3^2$$

$$W(\alpha_2, \alpha_3) = 2\alpha_3 - 4\alpha_2^2 + 4\alpha_2\alpha_3 - 2\alpha_3^2$$

To find the optimal values of α_1, α_2, and α_3, we compute the partial derivatives with respect to $\alpha_2 and \alpha_3$ and set them equal to zero. Then use their values to find α_1.

1. Partial derivative with respect to α_2:

$$\frac{\partial W}{\partial \alpha_2} = -8\alpha_2 + 4\alpha_3 = 0$$

$$\alpha_2 = \frac{\alpha_3}{2}$$

2. Partial derivative with respect to α_3:

$$\frac{\partial W}{\partial \alpha_3} = 2 + 4\alpha_2 - 4\alpha_3 = 0$$

$$2 + 4(\frac{\alpha_3}{2}) - 4\alpha_3 = 0$$

$$\alpha_3 = \frac{-2}{-2}$$

$$\alpha_3 = 1$$

Solving for α_2:

$$\alpha_2 = \frac{\alpha_3}{2}$$

$$\alpha_2 = \frac{1}{2}$$

Solving for α_1:

$$-\alpha_1 - \alpha_2 + \alpha_3 = 0$$
$$\alpha_1 = -\alpha_2 + \alpha_3$$
$$\alpha_1 = -\frac{1}{2} + 1$$
$$\alpha_1 = \frac{1}{2}$$

Thus, the optimal values are:

$$\alpha_1 = \frac{1}{2}, \quad \alpha_2 = \frac{1}{2}, \quad \alpha_3 = 1$$

Calculate the Weight Vector W

$$W = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\begin{aligned} W &= \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \alpha_3 y_3 x_3 \\ &= \frac{1}{2}(-1)(0,0) + \frac{1}{2}(-1)(2,2) + 1(1)(2,0) \\ &= (0,0) + (-1,-1) + (2,0) \\ W &= (1,-1) \end{aligned}$$

Calculate the term b(W_0)

Get any support vector as example:

$$b = y_i - W \cdot x_i$$

$$\begin{aligned} b &= y_3 - W \cdot x_3 \\ &= 1 - (1,-1) \cdot (2,0) \\ &= 1 - 2 + 0 \\ b &= -1 \end{aligned}$$

Thus, the optimal values are:

$$b = -1 \quad \textit{and} \quad W = (1,-1)$$

2. (20 points) In a separable case, when a multiplier $\alpha_i > 0$, its corresponding data point $(\boldsymbol{x}_i, y_i)$ is on the boundary of the optimal separating hyperplane with $y_i(\boldsymbol{w}^T \boldsymbol{x}_i) = 1$.

   [Hint: Consider a toy data set with two positive examples at ([0,0], +1) and ([1, 0], +1), and one negative example at ([0, 1], -1).] (Note that there will be three weights $\boldsymbol{w} = [w_0, w_1, w_2]$ by considering the bias.)

   Answer

In a separable case, a point (x_i, y_i) with $\alpha_i > 0$ is a support vector and lies on the margin boundary, satisfying $y_i(w^T x_i + w_0) = 1$. However, the inverse is not always true: a point can satisfy $y_i(w^T x_i + w_0) = 1$ even if $\alpha_i = 0$.

Example:

Given the dataset:

$$(0,0), 1$$
$$(1,0), 1$$
$$(0,1), -1$$

Let's derive the hyperplane:

1. **For point** $(0,0)$:
$$1 \cdot (w_1 \cdot 0 + w_2 \cdot 0 + w_0) = 1 \quad \Rightarrow \quad w_0 = 1$$

2. **For point** $(1,0)$:
$$1 \cdot (w_1 \cdot 1 + w_2 \cdot 0 + w_0) = 1 \quad \Rightarrow \quad w_1 + 1 = 1 \quad \Rightarrow \quad w_1 = 0$$

3. **For point** $(0,1)$:
$$-1 \cdot (w_1 \cdot 0 + w_2 \cdot 1 + w_0) = 1 \quad \Rightarrow \quad -1(w_2 + 1) = 1 \quad \Rightarrow \quad w_2 = -2$$

Thus, the hyperplane is:

$$w^T x + w_0 = 0 \quad \Rightarrow \quad -2x_2 + 1 = 0 \quad \Rightarrow \quad x_2 = 0.5$$

The point $(0,1)$ satisfies:

$$y_3(w^T x_3 + w_0) = -1(0 \cdot 0 - 2 \cdot 1 + 1) = -1(-1) = 1$$

This means $(0,1)$ is on the boundary $y_i(w^T x_i + w_0) = 1$ even though it is correctly classified and has $\alpha_3 = 0$.

Ultimately, this demonstrates that a point can be on the boundary $y_i(w^T x_i + w_0) = 1$ without having a positive Lagrange multiplier α_i.

3. (20 points) **Non-separable Case SVM:** In our lecture, we compared the hard-margin SVM and soft-margin SVM. Prove that the dual problem of soft-margin SVM is almost identical to the hard-margin SVM, except that α_is are now bounded by C (tradeoff parameter).

Proof:

1. Hard-Margin SVM Dual Problem:

The objective of hard-margin SVM is to maximize the margin while ensuring that all points are correctly classified, without allowing any misclassifications nor margin violations.

The primal problem for hard-margin SVM is:

$$\min_{w, w_0} \quad \frac{1}{2}||w||^2$$

subject to:

$$y_i(w^T x_i + w_0) \geq 1 \quad \forall i$$

The corresponding dual problem is:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

subject to:

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \forall i$$

2. Soft-Margin SVM Dual Problem:

The objective of soft-margin SVM is to maximize the margin while allowing for some misclassification. This is achieved by introducing slack variables ξ_i, which measure the amount of violation for each point.

The primal problem for soft-margin SVM is:

$$\min_{w, w_0, \xi} \quad \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i$$

subject to:

$$y_i(w^T x_i + w_0) \geq 1 - \xi_i \quad \forall i, \quad \xi_i \geq 0$$

The corresponding dual problem is:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

subject to:

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad \forall i$$

3. Comparison of Dual Problems:

- Hard-Margin SVM: The dual problem is constrained only by $\alpha_i \geq 0$.

- Soft-Margin SVM: The dual problem is constrained by $0 \leq \alpha_i \leq C$.

This additional constraint in the soft-margin SVM accounts for the tolerance to misclassifications. The parameter C controls the trade-off between maximizing the margin and minimizing classification errors.

This proves that the main difference lies in the upper bound on α_i introduced by the parameter C.

4. (20 points) **Kernel Function:** A function K computes $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\boldsymbol{x_i}^T \boldsymbol{x}_j$. Is this function a valid kernel function for SVM? Prove or disprove it.

Answer

To verify whether $K(x_i, x_j) = -x_i^T x_j$ is a valid kernel function, we first check if it satisfies the symmetry and positive semi-definiteness conditions.

Step 1: Check Symmetry

Definition: A kernel function $K(x_i, x_j)$ is symmetric if it satisfies:

$$K(x_i, x_j) = K(x_j, x_i) \quad \text{for all } x_i, x_j.$$

Symmetry Check:

For $K(x_i, x_j) = -x_i^T x_j$, then

$$K(x_i, x_j) = -x_i^T x_j$$

and

$$K(x_j, x_i) = -x_j^T x_i.$$

Result:

Since the dot product $x_i^T x_j$ is symmetric, meaning $x_i^T x_j = x_j^T x_i$, it follows that:

$$K(x_i, x_j) = -x_i^T x_j = -x_j^T x_i = K(x_j, x_i).$$

Thus, the function $K(x_i, x_j) = -x_i^T x_j$ is symmetric.

Step 2: Check Positive Semi-Definiteness Using a Counterexample

To determine if $K(x_i, x_j) = -x_i^T x_j$ is a valid kernel function, we need to verify if it satisfies the positive semi-definiteness condition. In this case, I will be using a simple counterexample.

Definition: A kernel matrix K is positive semi-definite if, for any non-zero vector $\mathbf{z}$, the quadratic form $\mathbf{z}^T K \mathbf{z} \geq 0$ holds.

Counterexample: Consider a simple case with two vectors:

$$x_1 = [1, 0], \quad x_2 = [0, 1]$$

Calculate the Kernel Matrix:

Using $K(x_i, x_j) = -x_i^T x_j$, the kernel matrix K for x_1 and x_2 is:

$$K = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) \\ K(x_2, x_1) & K(x_2, x_2) \end{bmatrix} = \begin{bmatrix} -x_1^T x_1 & -x_1^T x_2 \\ -x_2^T x_1 & -x_2^T x_2 \end{bmatrix}$$

Substituting the values:

$$K = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

Test Positive Semi-Definiteness

For K to be positive semi-definite, the quadratic form $\mathbf{z}^T K \mathbf{z} \geq 0$ must hold for any non-zero vector $\mathbf{z}$. Choose $\mathbf{z} = [1, 1]^T$:

$$\mathbf{z}^T K \mathbf{z} = [1, 1] \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = [1, 1] \begin{bmatrix} -1 \\ -1 \end{bmatrix} = -2$$

Since $\mathbf{z}^T K \mathbf{z} = -2$, which is negative, K is not positive semi-definite.

Final Conclusion

The function $K(x_i, x_j) = -x_i^T x_j$ is **not** a valid kernel for SVM because its kernel matrix can violate the positive semi-definiteness condition.

5. (20 points) **Support Vector Machine for Handwritten Digits Recognition**: You need to use the software package scikit-learn https://scikit-learn.org/stable/modules/svm.html to finish this assignment. We will use "svm.SVC()" to create a svm model. The handwritten digits files are in the "data" folder: train.txt and test.txt. The starting code is in the "code" folder. In the data file, each row is a data example. The first entry is the digit label ("1" or "5"), and the next 256 are grayscale values between -1 and 1. The 256 pixels correspond to a 16×16 image. You are expected to implement your solution based on the given codes. The only file you need to modify is the "solution.py" file. You can test your solution by running "main.py" file. Note that code is provided to compute a two-dimensional feature (symmetry and average intensity) from each digit image; that is, each digit image is represented by a two-dimensional vector. These features along with the corresponding labels should serve as inputs to your solution functions.

   (a) (5 points) Complete the **svm_with_diff_c()** function. In this function, you are asked to try different values of cost parameter c.

   (b) (10 points) Complete the **svm_with_diff_kernel()** function. In this function, you are asked to try different kernels (linear, polynomial and radial basis function kernels).

   (c) (5 points) Summarize your observations from (a) and (b) into a short report. In your report, please report the accuracy result and total support vector number of each model. A briefly analysis based on the results is also needed. For example, how the number of support vectors changes as parameter value changes and why.

Report

Effect of Cost Parameter C

Different values of C were used to train SVM models. The trained models were then applied to the testing dataset to evaluate their performance.

Cost C	**Accuracy**	**Support Vectors**
0.01	0.8939	1080
0.1	0.9623	414
1	0.9599	162
5	0.9623	104
10	0.9623	92

Table 1: Model Performance with Different Values of C

Analysis:

- **C = 0.01:** Achieved an accuracy of 89.39% using 1080 support vectors, indicating underfitting due to the large margin.
- **C = 0.1:** Accuracy significantly improved to 96.23% with 414 support vectors, showing that increasing C reduces the margin width, resulting in better performance.
- **C = 1:** Accuracy slightly decreased to 95.99%, with the number of support vectors dropping to 162, indicating a more precise decision boundary.
- **C = 5:** Accuracy returned to 96.23% using only 104 support vectors, demonstrating optimal performance with fewer support vectors.
- **C = 10:** Accuracy remained stable at 96.23% with only 92 support vectors, suggesting that further increases in C have minimal impact on performance but lead to a tighter margin.

Effect of Kernel Functions

The models were also trained using different kernel functions and evaluated on the testing dataset.

Kernel	Accuracy	Support Vectors
Linear	0.9599	162
Polynomial	0.9575	75
RBF	0.9623	90

Table 2: Model Performance with Different Kernel Functions

Analysis:

- **Linear Kernel:** Achieved an accuracy of 95.99% using 162 support vectors. The model performed well for linearly separable data but required a relatively high number of support vectors.
- **Polynomial Kernel:** Accuracy slightly decreased to 95.75%, while the number of support vectors dropped significantly to 75, suggesting that the polynomial kernel may have overfit the data.
- **RBF Kernel:** Achieved the highest accuracy of 96.23% with only 90 support vectors, indicating that the RBF kernel effectively handles non-linear boundaries with a good balance between accuracy and support vector count.