AI534 — Written Homework Assignment 2 (45 pts) —

This assignment covers Kernel methods and Support vector machines.

1. (Cubic Kernels.) (8 pts) In class, we showed that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$ was equivalent to mapping each $\mathbf{x} = (x_1, x_2) \in R^2$ into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$. What is the corresponding Φ function?

$K(x_i, x_j) = (x_i \cdot x_j + 1)^3 = (x_{1i}x_{1j} + x_{2i}x_{2j} + 1)^3$

$=(x_{1i}x_{1j})^3+3(x_{1i}x_{1j})^2(x_{2i}x_{2j})+3(x_{1i}x_{1j})(x_{2i}x_{2j})^2+(x_{2i}x_{2j})^3+3(x_{1i}x_{1j})^2+3(x_{2i}x_{2j})^2+6(x_{1i}x_{1j})(x_{2i}x_{2j})+3(x_{1i}x_{1j})+3(x_{2i}x_{2j})+1$

Thus, $\Phi(x) = \left(x_1^3, x_2^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, \sqrt{3}x_1^2, \sqrt{3}x_2^2, \sqrt{6}x_1x_2, \sqrt{3}x_1, \sqrt{3}x_2, 1\right)$

2. (Kernel or not). (5 pts) Suppose that K_1 and K_2 are kernels with feature maps ϕ_1 and ϕ_2 respectively. Is function $K(\mathbf{x}, \mathbf{z}) = c_1K_1(\mathbf{x}, \mathbf{z}) + c_2K_2(\mathbf{x}, \mathbf{z})$ for $c_1, c_2 > 0$ a kernel function? If your answer is yes, write down the corresponding ϕ in terms of ϕ_1 and ϕ_2. If not, provide a proof or explain why.

Mercer's Theorem: $K(\mathbf{x}, \mathbf{z})$ is a valid kernel if and only if for any finite samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, its corresponding kernel matrix is positive semi-definite

K_1 and K_2 are valid kernels, their kernel matrices are PSD. For any positive constants c_1 and c_2, the matrix $c_1K_1 + c_2K_2$ is also PSD. Thus, it is kernel function since it satisfies Mercer's condition.

Let's define ϕ :

$K_1(\mathbf{x}, \mathbf{z}) = \phi_1(\mathbf{x})^T\phi_1(\mathbf{z})$
$K_2(\mathbf{x}, \mathbf{z}) = \phi_2(\mathbf{x})^T\phi_2(\mathbf{z})$
Thus, $K(\mathbf{x}, \mathbf{z}) = c_1\phi_1(\mathbf{x})^T\phi_1(\mathbf{z}) + c_2\phi_2(\mathbf{x})^T\phi_2(\mathbf{z})$

By combining ϕ_1 and ϕ_2 we get:
$\phi(\mathbf{x}) = \begin{pmatrix} \sqrt{c_1}\phi_1(\mathbf{x}) \\ \sqrt{c_2}\phi_2(\mathbf{x}) \end{pmatrix}$.

Thus, $K(x, z)$can be expressed as:
$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T\phi(\mathbf{z}) = \begin{pmatrix} \sqrt{c_1}\phi_1(\mathbf{x}) \\ \sqrt{c_2}\phi_2(\mathbf{x}) \end{pmatrix}^T \begin{pmatrix} \sqrt{c_1}\phi_1(\mathbf{z}) \\ \sqrt{c_2}\phi_2(\mathbf{z}) \end{pmatrix}$.

3. Kernelizing Logistic Regression (10 pts) For this problem you will follow the example of kernelizing perceptron, to kernelize the logistic regression shown below.

Algorithm 1: Stochastic gradient descent for logistic regression

Input: $\{(\mathbf{x}_i, y_i)_{i=1}^N\}$(training data), γ(learning rate)
Output: learned weight vector $\mathbf{w}$

```
Initialize w = 0;
while not converged do
    for i = 1, ..., N do
        w ← w + γ(y_i − σ(w^T x_i))x_i
    end
end
```

Specifically, please:

(a) (4 pts) Argue that the solution $\mathbf{w}^*$ for logistic regression can be expressed as the weighted sum of training examples (similar to slide 8 of the kernel methods lecture)

Logistic regression update rule is:
$\mathbf{w} \leftarrow \mathbf{w} + \gamma \sum_{i=1}^{N} \left(y_i - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \mathbf{x}_i$.

Let $\alpha_i = \gamma \left(y_i - \sigma(\mathbf{w}^T \mathbf{x}_i)\right) \mathbf{x}_i$. *We can write this equation as:*
$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$.

Thus, we can express logistic regression as a weighted sum of training examples, where α_i *acts as the weight for each training example* $\mathbf{x}_i$.

(b) (6 pts) Modify the following stochastic gradient descent algorithm logistic regression algorithm to kernelize it. (Hint: similar to the bottom algorithm on slide 14 of the kernel method lecture, but instead of counter, you will learn a continuous weights for α's)

We can now express w *in terms of weighted sum of training example.*
$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$.

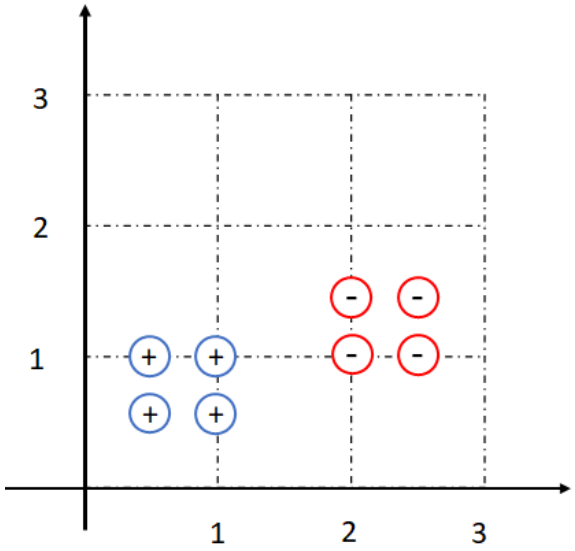
Where prediction $\hat{y} = w^T x$, *substituting* w *with* α *and* x*:*
$\hat{y} = \left(\sum_{i=1}^{N} \alpha_i \mathbf{x}_i\right)^T x$
$= \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^T \mathbf{x}$
$= \sum_{j=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x})$

Algorithm 2: Kernelized Stochastic Gradient Descent for Logistic Regression

Input: $\{(\mathbf{x}_i, y_i)_{i=1}^N\}$ *(training data),* γ *(learning rate)*
Output: *learned weight vector represented by* α

Initialize $\alpha = \mathbf{0}$;
while *not converged* **do**
  for $i = 1, \ldots, N$ **do**
    $\hat{y}_i \leftarrow \sum_{j=1}^{N} \alpha_j K(\mathbf{x}_j, \mathbf{x}_i)$;
    $\alpha_i \leftarrow \alpha_i + \gamma(y_i - \sigma(\hat{y}_i))$;
  end
end

4. (Hard margin SVM) (8 pts) Apply linear SVM without soft margin to the following problem.



a. (3pts) Please mark out the support vectors, the decision boundary ($w_1x_1 + w_2x_2 + b = 0$) and $w_1x_1 + w_2x_2 + b = 1$ and $w_1x_1 + w_2x_2 + b = -1$. You don't need to solve the optimization problem for this, you should be able to eyeball the solution and find the linear separator with the largest margin.

See figure 1

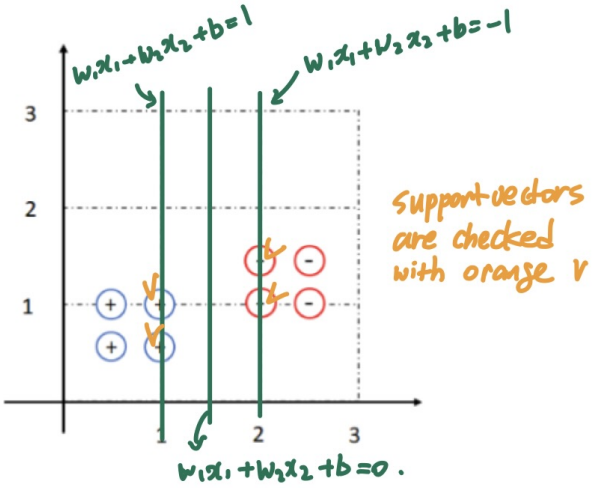


Figure 1:

b. (5 pts) Please solve for w_1, w_2 and b based on the support vectors you identified in (a). Hint: the support vectors would have functional margin = 1.

$w_1x_1 + w_2x_2 + b = 1$ *for positive class support vectors*
$w_1x_1 + w_2x_2 + b = -1$ *for negative class support vectors*

We can see that the positive class passes: $(1,1),(1,0.5)$
and the negative class passes : $(2,1),(2,1.5)$

Using this fact,
$w_1 \cdot 1 + w_2 \cdot 1 + b = 1$
$w_1 + w_2 + b = 1$

$w_1 \cdot 2 + w_2 \cdot 1 + b = -1$
$2w_1 + w_2 + b = -1$

$w_1 = -2, w_2 = 3 - b$

Now we can apply this to positive class support vector which passes $(1, 0.5)$
$-2 \cdot 1 + (3 - b) \cdot 0.5 + b = 1$
$0.5b = 1.5$
$b = 3, w_2 = 0$
Thus, $w_1 = -2, w_2 = 0, b = 3$

5. L_2 SVM (14 pts)
Given a set of training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $y_i \in \{1, -1\}$ for all i. The following is the primal formulation of L_2 SVM, a variant of the standard SVM obtained by squaring the slacks.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t. } & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,\ i \in \{1, \cdots, N\} \\ & \xi_i \geq 0,\ i \in \{1, \cdots, N\} \end{aligned}$$

a. (3pts) Show that removing the second constraint $\xi_i \geq 0$ will not change the solution to the problem. In other words, let $(\mathbf{w}^*, b^*, \xi^*)$ be the optimal solution to the problem without this set of constraints, show that $\xi_i^* \geq 0$ must be true, $\forall i \in \{1, \cdots, N\}$. (Hint: use proof by contradiction by assuming that there exists some $\xi_i^* < 0$.)
Proof by Contradiction.

Assumption:
Let's assume that there exists optimal solution $(\mathbf{w}^*, b^*, \xi^*)$ *when* $\xi_i^* < 0$, *if we don't have constraint* $\xi_i \geq 0$.

Contradiction:
i) Let's set $\xi_i^* = 0$. *Then we can modify objective function to* $\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w}$
ii) Let's see when $\xi_i^* < 0$. *The objective function is still* $\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{N}\xi_i^2$ *where* $c\sum_{i=1}^{N}\xi_i^2$ *is positive weight.*

This contradicts the assumption that there exists optimal solution $(\mathbf{w}^*, b^*, \xi^*)$ *when* $\xi_i^* < 0$, *if we don't have constraint* $\xi_i \geq 0$.
Since $c\sum_{i=1}^{N}\xi_i^2$ *is always positive and can not have smaller value than when* $\xi_i^* = 0$.

Thus, optimal solution to the problem without set of constraints, $\xi_i^* \geq 0$ *must be true,* $\forall i \in \{1, \cdots, N\}$.

b. (3 pts) After removing the second set of constraints, we have a simpler problem with only one set of constraints. Now provide the lagrangian of this new problem.
Lagrange multiplier $\alpha_i \geq 0$ *for each constraint* $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \geq 0$:

$\mathcal{L}(\mathbf{w}, b, \xi, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i)$.

c. (8pts) Derive the dual of this problem. How is it different from the standard SVM with hinge loss? Which formulation is more sensitive to outliers?

$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0$
$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$.

$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^N \alpha_i y_i = 0$
$\sum_{i=1}^N \alpha_i y_i = 0$.

$\frac{\partial \mathcal{L}}{\partial \xi_i} = 2c\xi_i - \alpha_i = 0$
$\xi_i = \frac{\alpha_i}{2c}$.

Substitute $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$ *and* $\xi_i = \frac{\alpha_i}{2c}$ *back into the Lagrangian*

Dual problem:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \frac{1}{4c}\sum_{i=1}^N \alpha_i^2,$$

$$s.t. \sum_{i=1}^N \alpha_i y_i = 0 \quad and \quad \alpha_i \geq 0.$$

Standard SVM uses a linear penalty, while dual formulation penalizes the square of the slack variables $\sum_i \xi_i^2$.
This makes Standard SVM is generally less sensitive to outliers than the hinge-loss SVM because squaring large slacks amplifies the penalty, discouraging large deviations more aggressively than the linear penalty in hinge loss.