# Machine Learning Assignment 1

Krunal Rathod

October 26, 2023

## 1 Linear Models and Kernel Methods

### 1.1 Problem 1. Ridge Regression (15 Points)

Given the loss function in ridge regression:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\left((f(\varphi(x_n)) - t_n)^2 + \frac{\lambda}{2}\|w\|^2\right)$$

We want to find the optimal parameters $w$ that minimize this loss function.
1. Take the derivative of the loss function with respect to $w$:

$$\nabla E(w) = \sum_{n=1}^{N}((f(\varphi(x_n)) - t_n)\nabla f(\varphi(x_n)) + \lambda w)$$

2. Since $f(x) = w^T\varphi(x)$, $\nabla f(\varphi(x_n)) = \varphi(x_n)$, so we have:

$$\nabla E(w) = \sum_{n=1}^{N}\big((w^T\varphi(x_n) - t_n)\varphi(x_n) + \lambda w\big)$$

3. Set the gradient equal to zero to find the optimal parameters:

$$\sum_{n=1}^{N}\big((w^T\varphi(x_n) - t_n)\varphi(x_n) + \lambda w\big) = 0$$

4. Rearrange the terms to isolate $w$:

$$\sum_{n=1}^{N}(w^T\varphi(x_n) - t_n)\varphi(x_n) + \lambda w = 0$$

5. Rewrite this as a matrix equation:

$$\left(\sum_{n=1}^{N}\varphi(x_n)\varphi(x_n)^T + \lambda I\right)w = \sum_{n=1}^{N}t_n\varphi(x_n)$$

6. Solve for $w$:

$$w = \left(\sum_{n=1}^{N}\varphi(x_n)\varphi(x_n)^T + \lambda I\right)^{-1}\sum_{n=1}^{N}t_n\varphi(x_n)$$

This is the closed-form solution for the optimal parameters $w$ in ridge regression, and it takes into account both the data and the regularization term $\lambda$.

### 1.2 Problem 2. Feature Engineering (10 points)

To achieve linear separability between classes C1 and C2 in a 2D point set $S$, we can use radial basis functions (RBF). Here's the propose new features:

1. **Calculate RBF (Radial Basis Functions):** - For class C1 (blue points), calculate the Euclidean distance from the origin (0,0) to each point $(x(1), x(2))$. We can use this distance as the RBF value. - For class C2 (yellow points), calculate the distance from the origin (0,0) to each point $(x(1), x(2))$, but subtract a smaller constant than previous one to differentiate them from class C1.
2. **Define New Features:** - For each point $(x(1), x(2))$, the calculated RBF values will now serve as the new features. These new features create a transformed feature space, which looks like a ripple pattern.
3. **Mathematical Representation of New Features:**
   - For class C1:
   $$f_1(x(1), x(2)) = e^{-(x(1)^2+x(2)^2-2^2)}$$
   - For class C2:
   $$f_2(x(1), x(2)) = e^{-(x(1)^2+x(2)^2-c^2)}$$

In these equations: - $x(1)$ and $x(2)$ represent the original coordinates of a point. - $f_1(x(1), x(2))$ and $f_2(x(1), x(2))$ represent the new features for classes C1 and C2, respectively.

- The RBF values are calculated based on the Euclidean distance from the origin (0, 0) to the point $(x(1), x(2))$.

- The constants 2 and $c$ are used to control the influence of the RBF values, which creates the pattern of the ripple for these data points.

By applying these RBF transformations to the original 2D points, we create a new feature space where classes C1 and C2 should be linearly separable, allowing for the linear classifiers or decision boundaries to distinguish between the two classes.

### 1.3 Problem 3. Kernel Function (12 Points)

To check whether the given function $f(x, y) = x^T \cdot x \cdot x^T \cdot y \cdot y^T \cdot y$ is a valid kernel according to the provided rules for valid kernels, we can express $f(x, y)$ in a form that adheres to the rules.

The provided rules for valid kernels include the rule $k(x, y) = x^T \cdot Ay$, where $A$ is a symmetric positive semi definite matrix. To determine whether $f(x, y)$ can be expressed in this form, we need to find an appropriate $A$ such that:

$$f(x, y) = x^T \cdot x \cdot x^T \cdot y \cdot y^T \cdot y = x^T \cdot Ay$$

To simplify this expression and identify $A$, let's examine the structure of $f(x, y)$. The expression $x^T \cdot x$ is the inner product of vector $x$ with itself, and $y \cdot y^T$ is the outer product of vector $y$ with itself. If we define $A$ as the product of these terms, we have:

$$A = x^T \cdot x \cdot y \cdot y^T$$

Now, we can express $f(x, y)\ as:$

$$f(x, y) = x^T \cdot Ay$$

With this definition of $A$, $f(x, y)$ can be expressed in the form required by the provided rules. Furthermore, $A$ can be a symmetric positive semi definite matrix because it is derived from inner and outer products of $x$ and $y$ respectively.

Therefore, based on the provided rules for valid kernels, the function $f(x, y) = x^T \cdot x \cdot x^T \cdot y \cdot y^T \cdot y$ can be considered a valid kernel.

### 1.4 Problem 4. SVM (15 Points)

1. **Plot of Training Points and Linear Separability**: - When we plot the six training points, we observe that the classes {+, -} are linearly separable. This means that it's possible to draw a straight line (hyperplane) that separates the two classes without any points from one class spilling over to the other side.

2. **Maximum Margin Hyperplane and Support Vectors**: By examining the dataset and the pattern of data points, we can say that the hyperplane has a slope of -1. To maximize the margin, the hyperplane should pass through the point where the positive class is closest to the negative class, which in this case is the point (1, 1) from the positive class.

So, the equation of the hyperplane becomes:

$$x_1 + x_2 - b = 0$$

To calculate the weight vector $\mathbf{w} = [w_1, w_2]$, we use the coefficients of $x_1$ and $x_2$ in the equation:

$$w_1 = 1 \qquad\qquad w_2 = 1$$

The bias term $b$ can be calculated using any of the closest support vectors. Let's use the closest support vector (1, 1):

$$1 + 1 - b = 0$$

$$b = 2$$

So, the equation of the maximum margin hyperplane is:

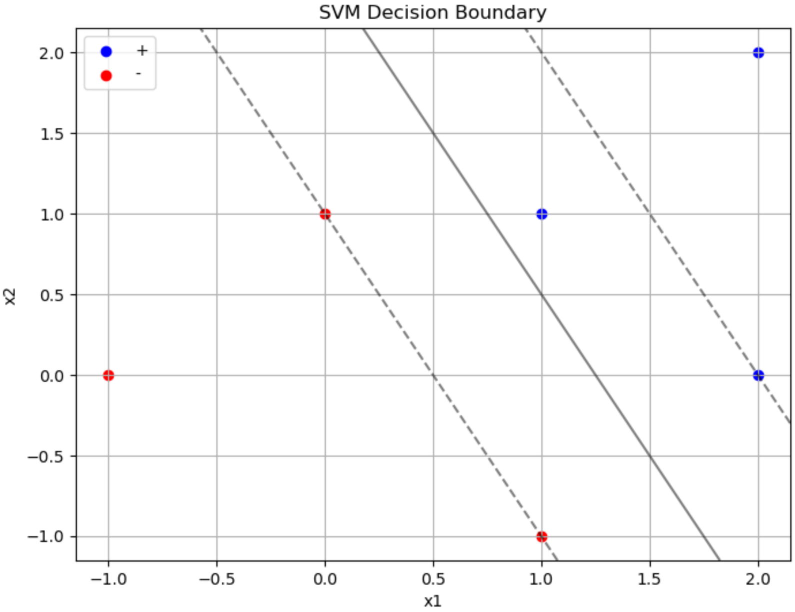
$$x_1 + x_2 - 2 = 0$$

Now, the support vectors are the data points that are closest to the hyperplane. For this dataset, the support vectors are as follows:

For the positive class: - Support Vector 1: (1, 1) - Support Vector 2: (2, 0)

For the negative class: - Support Vector 3: (1, -1) - Support Vector 4: (0, 1)

These points are the support vectors as they are closest to the maximum margin hyperplane.



3. **Effect of Removing Support Vectors on Margin**: If one of the support vectors near the decision boundary is removed, the margin size may decrease because the nearest support vector is no longer available to help maximize the margin.

4. **Is your answer to (3) also true for any dataset? Provide a counterexample or give a short proof.**

Counterexample:

Suppose we have a dataset with four data points, arranged as follows:

**Class** $+$:

1. Support Vector 1: $(1, 1)$
2. Support Vector 2: $(2, 2)$

**Class** $-$:

1. Non-Support Vector: $(-1, -1)$
2. Support Vector 4: $(0, 0)$

In this scenario, all four data points are support vectors as they contribute to defining the decision boundary. The margin size is determined by the distance between the decision boundary and the nearest support vectors. If we remove Support Vector 3, the margin size will be determined by Support Vectors 1, 2, and 4, which are now the closest points to the decision boundary. The margin size will likely be larger with all four support vectors compared to when Support Vector 3 is removed. This proves that the removal of a support vector can result in a situation where the margin size decreases.