INTRODUCTION TO MACHINE LEARNING
Assignment 3
Kernels, SVM

Deadline: November 12th, 11:59 p.m. IST **Max points: 100**

1. **(Ridge regression and the kernel trick)** Consider a regression model with the following cost function instead of the usual one

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\theta^T x^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}\sum_{j=0}^{m}\theta_j^2,$$

where $\{x^{(i)} \in \mathbb{R}^{m+1}, y^{(i)} \in \mathbb{R}, i = 1, ..., n\}$ are the training data, $\theta = \begin{bmatrix}\theta_0 & \cdots & \theta_m\end{bmatrix}^T$, and $\lambda > 0$.

   (a) Let us use X to denote the matrix whose ith row is $x^{(i)}$, y to denote the column vector of $y^{(i)}$'s. Then we know that $\frac{1}{2}\sum_{i=1}^{n}(\theta^T x^{(i)} - y^{(i)})^2$ is minimized when $\hat{\theta} = (X^T X)^{-1} X^T y$. Derive a similar-looking closed-form expression for $\hat{\theta}$ that minimizes $J(\theta)$.

   (b) Prove that subject to the existence of products and inverses, for matrices A and B, $(\lambda I + BA)^{-1}B = B(\lambda I + AB)^{-1}$.
   (Hint: What would this statement look like if the inverses were taken to the other sides?)

   (c) Suppose now that we want to use kernels to represent our m features in a higher dimensional space. For the feature map ϕ, our cost function becomes

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(\theta^T \phi(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2}\sum_{j=0}^{m}\theta_j^2.$$

   For a new input x', the output will now be $\theta^T\phi(x')$. The kernel trick makes it possible to make a prediction for x' without computing the high-dimensional dot product of $\theta^T\phi(x')$. Show how this can be done. You may assume that θ is expressible as a linear combination of $\phi(x^{(i)})$'s (i.e., $\theta = \sum_{i=1}^{n}\alpha_i\phi(x^{(i)})$ for some parameters α_i).
   ***(Hint:** use the identity from part (b))*

(6 + 3 + 6 = 15)

2. **(Alternative soft margin SVM))** If the given data $\{x^{(i)} \in \mathbb{R}^m\}$ are not linearly separable, then we should modify the support vector machine algorithm by introducing and error margin that is then minimized ($C\sum_{i=1}^{n}\xi_i$, $\xi_i \geq 0$). Suppose we instead consider the following minimization problem:

$$\min_{w_i,b,\xi} \frac{1}{2}\sum_{i=1}^{m} w_i^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2$$
$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \; i = 1, ..., n.$$

   (a) Show that the optimal solution no longer requires the constraints of $\xi_i \geq 0 \; \forall \; i$.

   (b) Formulate the Lagrangian for this problem, and minimize it by computing the necessary gradients.

   (c) Formulate the dual of the given minimization problem.

(3 + 6 + 6 = 15)

3. **(Kernel or not?)** Show that

(a) $k(x,z) = (xz+1)^2$ is/isn't a valid kernel.

(b) $k(x,z) = (xz-1)^3$ is/isn't a valid kernel.

(4 + 4 = 8)

4. **(SVM by hand)** Consider a dataset with the following data points, where $y^{(i)}$ represent the labels:

$$\{(x^{(i)}, y^{(i)}\}_i = \{(-3,+1), (-2,+1), (-1,-1), (0,-1), (1,-1), (2,+1), (3,+1)\}$$

Consider mapping the $x^{(i)}$'s to 2 dimensions, using the feature map $\phi(x) = (x, x^2)$, and the minimization problem

$$\min_{w_1, w_2, b} w_1^2 + w_2^2$$
$$\text{s.t. } y^{(i)}(w_1 x_1^{(i)} + w_2 x_1^{(i)2} + b) \geq 1 \ \forall\ i.$$

(a) Plot the given data in $\mathbb{R}^2$ and draw the decision boundary of the max margin classifier.

(b) What is the value of the margin achieved by the optimal decision boundary?

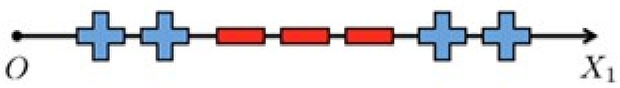
(c) What is a vector that is orthogonal to the decision boundary?

(d) What are the support vectors of the classifier?

(6 + 3 + 3 + 3 = 15)

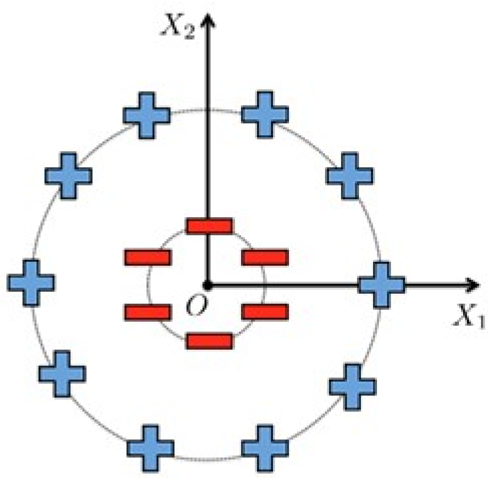
5. **(Transformations and separability)**

(a) Consider the following one dimensional dataset, and suggest a 1D transformation to make the points linearly separable.

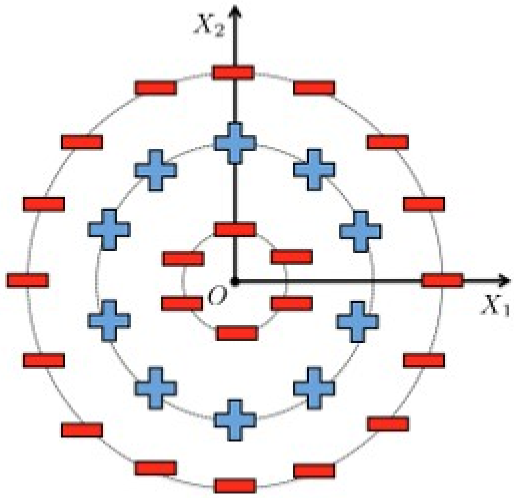


(b) For the same dataset, what can be a 2D transformation to make the points linearly separable?

(c) What is a 1D transformation to make the following data points linearly separable?



(d) Use ideas from the above datasets to suggest a 2D transformation that makes the following data points linearly separable.



(3 + 3 + 3 + 3 = 12)

6. **(Programming - Non-linear decision boundaries without kernels)**

   (a) Generate a two-dimensional dataset $\{(x_1^{(i)}, x_2^{(i)}), i = 1, ..., 500\}$ where $x_1^{(i)}, x_2^{(i)}$ follow uniform (0,1). Assign them two classes based on whether or not $x_1^2 > x_2^2$.

   (b) Plot the data points, colour coded as per the classes.

   (c) Fit a logistic regression model on the data, using x_1 and x_2 as features.

   (d) Apply this model to the data to obtain predicted class labels for each point. Plot the points, colour coded by *predicted* class labels.

   (e) Now, fit a logistic regression model to the data using non-linear functions of x_1 and x_2. Try x_1^2, x_2^2, x_1x_2, $\log(x_1)$, $\log(x_2)$.

   (f) Repeat step (d) with the new model.

   (g) Use x_1 and x_2 as features to fit a support vector classifier to the data, use the classifier to make predictions on the data points, and plot the points colour coded as per the predictions of this model.

   (h) Repeat the above step for a support vector machine that uses a non-linear kernel.

   (i) Report your observations for each classifier.

(5 + 3 + 5 + 3 + 6 + 3 + 5 + 3 + 2 = 35)