Kernel Methods

Instructions:

- Do not change the structure of the `/Code/` directory and place your code and the outputs in appropriate directories.
- A single report, containing containing all the plots in the correct sequence along with any other details must be submitted separately in the PDF format.

Question 1: Valid Kernels

a) Derive the necessary and sufficient conditions on matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ needed to ensure that the function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal \mathbf{A} \mathbf{y}$ is a valid kernel. Here, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ are d-dimensional vectors.

(5 points)

b) Install the dill package [`https://pypi.org/project/dill/`] in Python[1]. You are given five serialized dumps of functions in `k1.pkl`, `k2.pkl`, `k3.pkl`, `k4.pkl`, and `k5.pkl` in the `/Code/Q1` directory. We need to identify which of the above mentioned functions are likely to implement a valid kernel. As the functional form is not known, we will resort to empirical validation[2]. To do so, compute a matrix $K \in \mathbf{R}^{n \times n}$ using the following procedure:

Load k_ℓ from the appropriate `.pkl` file
for $i = 1 \to n$ **do**
  Sample $\mathbf{x}^{(i)} \sim \text{domain}(k_\ell)$ from the domain of k_ℓ
end for
for $i = 1 \to n$ **do**
  for $j = 1 \to n$ **do**
    Compute $K_{ij} = k_\ell(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$
  end for
end for

Answer the following questions using the procedure mentioned above. Use $n = 100$ for all sub-parts. You can be more confident about your answer either by choosing a larger value of n or by independently running the experiment several times. For part i) - iv), sample each component of $\mathbf{x}^{(i)} \in \mathbb{R}^3$ independently from the uniform distribution on $[-5, +5]$.

i) Is $k_1 : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ implemented in `k1.pkl` likely to be a valid kernel?

(3 points)

ii) Is $k_2 : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ implemented in `k2.pkl` likely to be a valid kernel?

(3 points)

iii) Is $k_3 : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ implemented in `k3.pkl` likely to be a valid kernel?

[1] Tested on Python 3.7

[2] Note that empirical validation alone is not enough to give a conclusive answer when analytical form of the function is unknown. Nonetheless, in certain cases, it can be very quickly checked that the function is not a kernel by randomly evaluating it at many points. In this question, the larger the value of n you choose, the more confident you will be in your answer.

(3 points)

iv) Is $k_4 : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ implemented in `k4.pkl` likely to be a valid kernel?

(3 points)

v) Load $k_5 : \mathcal{C}_3 \times \mathcal{C}_3 \to \mathbb{R}$ from `k2.pkl` where $\mathcal{C}_3 = \{\mathbf{x} \in [0, \infty)^3 : ||\mathbf{x}||_2^2 = 1\}$. To sample $\mathbf{x}^{(i)} \in \mathcal{C}_3$, call the function `k5sampler()` implemented in file `/Code/Q1/k5sampler.pkl`. This function does not take any arguments and returns a 3-dimensional vector in the domain of k_5 as output. Is k_5 likely to implement a valid kernel?

(3 points)

Question 2: Support Vector Machines

a) Sample $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{2\times 2})$ and $b \sim \mathcal{N}(0, 1)$ where $\mathbf{I}_{2\times 2}$ is an identity matrix of size 2×2. Sample $n = 100$ data points $\mathbf{x}^{(i)} \in \mathbb{R}^2$ i.i.d. by independently sampling each component of $\mathbf{x}^{(i)}$ from a uniform distribution on $[-3, +3]$, for $i = 1, 2, \ldots, n$. Compute the label $y^{(i)} \in \{-1, +1\}$ for data point $\mathbf{x}^{(i)}$ as $y^{(i)} = \text{sign}(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b)$. Plot the generated data using a scatter plot and color the points based on the labels. Submit the generated plot.

(3 points)

b) The primal formulation of SVMs[3] for the case where the data may not be linearly separable is given by:

$$\begin{aligned} \min_{\xi, \mathbf{w}, b} \ & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t. } & y^{(i)}(\mathbf{w}^\intercal \mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \ i = 1, 2, \ldots, n \\ & \xi_i \geq 0, \ i = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

Here, $\xi = [\xi_1, \xi_2, \ldots, \xi_n]$ is a n-dimensional vector of slack variables that are constrained to be non-negative, $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ are the parameters specifying the separating hyperplane, and C is a user specified hyper-parameter. Given the training set of $n = 100$ data points generated in Question **2(a)**, write code to solve the primal problem given in equation (1)[4]. Tune the value of hyper-parameter C to obtain the best performance on the training set. Sample another 50 points as in Question **2(a)** and treat them as test set. Report the accuracy on training and test set. Also report the optimal value of C, $\mathbf{w}$ and b. Note that $d = 2$ in this question.

(7 points)

c) We will now use a different method for labeling the data points. As in Question **2(a)**, start by sampling $n = 100$ data points $\mathbf{x}^{(i)} \in \mathbb{R}^2$, for $i = 1, 2, \ldots, n$ by independently sampling each component of $\mathbf{x}^{(i)}$ from a uniform distribution over $[-3, +3]$. Now, compute the label $y^{(i)} \in \{-1, +1\}$ as follows:

$$y^{(i)} = \begin{cases} +1, & \text{if } (x_1^{(i)})^2 + \frac{(x_2^{(i)})^2}{2} \leq 2 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

i) Plot the generated data as a scatter plot. Use different colors for different classes.

(2 points)

[3] For reference, see: `http://cs229.stanford.edu/notes/cs229-notes3.pdf`

[4] You can use the *cvxopt* package in Python to solve the quadratic program

ii) Use your code from Question **2(b)** to find a separating hyperplane for the training set consisting of the newly generated data. As before, sample another 50 points, and label them using equation (2). Treat these points as test set. Tune the paramter C to obtain the best performance on training set. Report the optimal value of C, $\mathbf{w}$ and b along with the accuracy on the training and test set.

(3 points)

d) Define a mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ as follows:

$$\phi([x_1, x_2]) = [x_1^2, x_2^2]. \tag{3}$$

Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ be the data generated in Question **2(c)**. Obtain a modified dataset $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{x}}^{(i)}, y^{(i)})\}_{i=1}^n$, where $\tilde{\mathbf{x}}^{(i)} = \phi(\mathbf{x}^{(i)})$ and the label $y^{(i)}$ has been copied from the original dataset $\mathcal{D}$.

i) Plot the generated data as a scatter plot. Use different colors for different classes.

(2 points)

ii) Use your code from Question **2(b)** to find a separating hyperplane for the training set $\tilde{\mathcal{D}}$. Sample another 50 points, and label them using equation (2). Obtain the modified dataset as above by applying the mapping ϕ. Treat these points as test set. Tune the paramter C to obtain the best performance on training set. Report the optimal value of C, $\mathbf{w}$ and b along with the accuracy on the training and test set.

(3 points)

iii) Is the performance reported in the previous question better than that reported in Question **2(c)**? Explain your observation in a sentence.

(2 points)

e) Write code to solve the dual of the primal problem in equation (1)[5]. Write an appropriate kernel function to optimize the performance on the training data generated in Question **2(c)**. Provide the expression for the kernel function used by you in your report. Also report the best training and test accuracy as before along with any hyper-parameters that were used by you.

(8 points)

Question 3: Kernelized-Regression

a) Generate $n = 100$ data points $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}$ and $y_i \in \mathbb{R}$, by sampling $x_i \sim \text{unif}[-1, 1]$ and setting $y_i = \sin(3x_i)$. Plot the data and save the file in the `/Code/Q3/` directory.

(2 points)

b) Fit a linear regression model with parameters $w, b \in \mathbb{R}$ to the data from Question **3(a)**. Plot the learned line along with the training data. Use different color for the line and for the data points. Report the mean square error on the training set.

(5 points)

c) For $k = 1, 2, \ldots, 10$, define $\phi_k : \mathbb{R} \to \mathbb{R}^k$ as $\phi_k(x) = [x^1, x^2, \ldots, x^k]$. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the dataset that was sampled in Question **3(a)** and let $\mathcal{D}_k = \{(\phi_k(x_i), y_i)\}_{i=1}^n$, where x_i, y_i are from $\mathcal{D}$. Follow the pseudocode:

for $k = 1 \to 10$ **do**
  Train a linear regression model on $\mathcal{D}_k$
  Report the mean squared error on $\mathcal{D}_k$

[5] As before, you can use the *cvxopt* package to solve the quadratic program.

Plot the learned function $f(x) = \mathbf{w}^\intercal \phi_k(x) + b$ and training data $\mathcal{D}$ in the same plot[6]
end for

(5 points)

d) Implement kernelized linear regression (Section 6.1, Pattern Recognition and Machine Learning - Bishop, 2006). Use the data from Question **3(a)** as training data. Choose an appropriate kernel function to optimize the performance on training set. Sample a test set of $n = 100$ points as in Question **3(a)**. Do the following: **(i)** specify the kernel used by you, **(ii)** report training set and test set mean-squared error, and **(iii)** plot the learned function along with training data.

(8 points)

Question 4: Kernel K-Means

Let $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^n$ be a dataset containing n examples, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ represents the i^{th} example, and let $\phi : \mathbb{R}^d \to \mathbb{R}^m$ be a function that maps the input features to a high dimensional space (usually $m >> d$). The K-means clustering algorithm requires the computation of distance between data points and the mean feature vector of each cluster[7]. Briefly, for K clusters, the algorithm can be summarized as:

Randomly initialize $\mu_k \in \mathbb{R}^d$, for $k = 1, 2, \ldots, K$
while not converged **do**
  $c^{(i)} \leftarrow \text{argmin}_{k=1,2,\ldots,K} ||\mathbf{x}^{(i)} - \mu_k||_2^2$, for $i = 1, 2, \ldots, n$
  $\mu_k = \frac{1}{|\{i:c^{(i)}=k\}|} \sum_{i:c^{(i)}=k} \mathbf{x}^{(i)}$
end while

a) We need to compute the distance between $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. However, we do not have access to the mapping ϕ, but only have access to a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that computes $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\intercal \phi(\mathbf{y})$. Write a function `dist(x, y, k)` that takes the kernel function k as input along with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and computes the distance between $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$. Use the kernel function given in `/Code/Q4/kernel_4a.pkl`. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d \in \{0, 1\}^d$ be such that $\mathbf{e}_i$ has its i^{th} entry set to one and very other entry set to zero. Compute the distance matrix $D \in \mathbb{R}^{d \times d}$ such that D_{ij} contains the distance between $\phi(\mathbf{e}_i)$ and $\phi(\mathbf{e}_j)$. Report the sum of all entries of D. Use $d = 10$.

(5 points)

b) Write a function to compute the distance between $\phi(\mathbf{x})$ and $\mu = \frac{1}{m} \sum_{i=1}^m \phi(\mathbf{y}_i)$ where $\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_m \in \mathbb{R}^d$. With notation borrowed from Question **4(a)**, let d_i be the distance between $\phi(\mathbf{e}_i)$ and $\frac{1}{d} \sum_{i=1}^d \phi(\mathbf{e}_i)$. Report the value of $\sum_{i=1}^d d_i$. Use $d = 10$.

(5 points)

c) Use the code from Question **4(b)** to implement the kernelized K-means clustering algorithm which is essentially the K-means clustering algorithm, but executed in the high dimensional space on the data $\tilde{\mathcal{D}} = \{\phi(\mathbf{x}^{(i)})\}_{i=1}^n$ where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ for $i = 1, 2, \ldots, n$. Use the kernel function given in `/Code/Q4/kernal_4a.pkl` and load the data from `/Code/Q4/data.npy`. The data consists of $n = 500$, $d = 2$ dimensional data points. Set $K = 5$, plot the data points on a scatter plot and color them based on cluster membership.

(5 points)

[6]Note that $f(x)$ is a function in one variable
[7]For details see: `http://cs229.stanford.edu/notes/cs229-notes7a.pdf`

Question 5: Kernel Fisher's Discriminant Analysis

Let $\mathcal{C}_1 = \{\mathbf{x}^{(i)}\}_{i=1}^{n_1} \subset \mathbb{R}^d$ and $\mathcal{C}_2 = \{\mathbf{x}^{(j)}\}_{j=1}^{n_2} \subset \mathbb{R}^d$ be sets corresponding to data points in classes C_1 and C_2 respectively. Fisher's Linear Discriminant Analysis finds a low dimensional representation of data points (Section 4.1.4, Pattern Recognition and Machine Learning - Bishop, 2006). Let $x^{(i)} = \mathbf{w}^\intercal \mathbf{x}^{(i)}$ be the scalar corresponding to the projection of data point $\mathbf{x}^{(i)}$ on the direction specified by $\mathbf{w}$. The objective is to find $\mathbf{w}$ such that: **(i)** $(\mu_1 - \mu_2)$ is maximized where $\mu_k = \frac{1}{n_k} \sum_{\mathbf{x} \in \mathcal{C}_k} \mathbf{x}$ for $k = 1, 2$, and **(ii)** variance of projected points within a class in minimized. To find $\mathbf{w}$, one minimizes the following cost function:

$$J(\mathbf{w}) = \frac{\mathbf{w}^\intercal \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\intercal \mathbf{S}_W \mathbf{w}}, \tag{4}$$

where,

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\intercal$$
$$\mathbf{S}_W = \sum_{\mathbf{x} \in \mathcal{C}_1} (\mathbf{x} - \mathbf{m}_1)(\mathbf{x} - \mathbf{m}_1)^\intercal + \sum_{\mathbf{x} \in \mathcal{C}_2} (\mathbf{x} - \mathbf{m}_2)(\mathbf{x} - \mathbf{m}_2)^\intercal.$$

Here, $\mathbf{m}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in \mathcal{C}_k} \mathbf{x}$ for $k = 1, 2$, and $\mathbf{S}_B$ and $\mathbf{S}_W$ are the *between-class* and *within-class* covariance matrices respectively. One can show that the optimal solution of $\mathbf{w}$ is given by:

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2). \tag{5}$$

Note that we are just interested in the direction $\mathbf{w}$ so the exact magnitude of $\mathbf{w}$ does not matter. Fisher's Linear Discriminant Analysis (LDA) looks for a linear projection of data and, in some cases, such a linear projection might yield poor results. However, we can first map the data into a high-dimensional space and then perform Fisher's LDA in that space. This is the basic idea behind Kernel Fisher's Discriminant Analysis. Let $\phi : \mathbb{R}^d \to \mathbb{R}^m$ be a function that maps the examples to a high dimensional feature space. We want to execute Fisher's LDA on the examples in $\tilde{\mathcal{C}}_1 = \{\phi(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}_1}$ and $\tilde{\mathcal{C}}_2 = \{\phi(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}_2}$. As in the case of kernel K-means, in general, ϕ is not known, but a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is given such that $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\intercal \phi(\mathbf{y})$.

a) Generate data points as follows: $\mathcal{C}_1 = \{x^{(i)} \sim \text{unif}[-1, +1]\}_{i=1}^{n_1}$ and $\mathcal{C}_2 = \{x^{(j)} \sim \text{unif}([-3, -2] \cup [2, 3])\}_{j=1}^{n_2}$, where all the sampled data points are real numbers, $n_1 = n_2 = 100$. Plot the data as a scatter plot using different colors for different classes.

(2 points)

b) Define $\phi : \mathbb{R} \to \mathbb{R}^2$ as $\phi(x) = [x, x^2]$. Plot the data $\{\phi(x)\}_{x \in \mathcal{C}_1 \cup \mathcal{C}_2}$ as a scatter plot using different colors for different classes.

(2 points)

c) Let the data generated in Question **5(a)** be $\mathcal{C}_1 = \{x^{(i)}\}_{i=1}^{n_1}$ and $\mathcal{C}_2 = \{x^{(j)}\}_{j=1}^{n_2}$. Further, let $\tilde{\mathcal{C}}_1 = \{\phi(x)\}_{x \in \mathcal{C}_1}$ and $\tilde{\mathcal{C}}_2 = \{\phi(x)\}_{x \in \mathcal{C}_2}$, where ϕ is defined as in Question **5(b)**. Implement Fisher's Linear Discriminant Analysis on data given in $\tilde{\mathcal{C}}_1$ and $\tilde{\mathcal{C}}_2$. Plot the points in $\tilde{\mathcal{C}}_1$ and $\tilde{\mathcal{C}}_2$ using different colors. On the same plot, draw the line corresponding to the learned direction $\mathbf{w} \in \mathbb{R}^2$.

(6 points)

d) Implement Kernel Fisher's Discriminant Analysis (Section 5.4, Kernel Methods for Pattern Analysis - Shawe-Taylor and Cristianni). Use the data from Question **5(a)**, obtain a one-dimensional representation of data, and choose a threshold θ to classify the points into two classes. Choose an appropriate kernel function that maximizes the classification performance. Report the kernel function used by you along with the classification accuracy. Plot the one-dimensional representation of data as a scatter plot and color points based on true class memberships. Indicate θ on this plot. In Kernel Fisher's Discriminant Analysis the weight vector $\mathbf{w} = \sum_{x \in \mathcal{C}_1 \cup \mathcal{C}_2} \alpha_x \phi(x)$. Plot the learned α_x on y-axis against $x \in \mathcal{C}_1 \cup \mathcal{C}_2$.

(10 points)