Q5. [6 pts] Kernels and Feature Transforms

A kernel function $K(x, z)$ is a function that conceptually denotes the similarity between two instances x and z in a transformed space. More specifically, for a feature transform $x \rightarrow \phi(x)$, the kernel function is $K(x, z) = \phi(x) \cdot \phi(z)$. The beauty of algorithms using kernel functions is that we never actually need to explicitly specify this feature transform $\phi(x)$ but only the values $K(x, z)$ for pairs (x, z). In this problem, we will explore some kernel functions and their feature transforms. For this problem the input vectors are assumed to be 2 dimensional (i.e. $x = (x_1, x_2)$). Remember that $x \cdot z = x_1 z_1 + x_2 z_2$.

(a) For each of the kernel functions below, mark the corresponding feature transform: (mark a single option only for each question)

(i) [1 pt] $K(x, z) = 1 + x \cdot z$

- ○ $\phi(x) = (x_1, x_2)$
- ○ $\phi(x) = (1, x_1, x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (x_1^2, x_2^2)$
- ○ $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2)$

(ii) [1 pt] $K(x, z) = (x \cdot z)^2$

- ○ $\phi(x) = (x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$

(iii) [1 pt] $K(x, z) = (1 + x \cdot z)^2$

- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, x_1, x_2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1, x_2, \sqrt{2}x_1x_2)$
- ○ $\phi(x) = (1, x_1x_2, x_1^2x_2^2)$

(b) Multiple kernels can be combined to produce new kernel functions. For example $K(x, z) = K_1(x, z) + K_2(x, z)$ is a valid kernel function. For the questions below, kernel K_1 has the associated feature transform ϕ_1 and similarly K_2 has the feature transform ϕ_2. Mark the feature transform associated with K for the expressions given below.

Note: The operator $[*, *]$ denotes concatenation of the two arguments. For example, $[x, z] = (x_1, x_2, z_1, z_2)$.

(i) [1 pt] $K(x, z) = aK_1(x, z)$, for some scalar $a > 0$

- ○ $\phi(x) = \phi_1(x)$
- ○ $\phi(x) = [a, \phi_1(x)]$
- ○ $\phi(x) = a\phi_1(x)$
- ○ $\phi(x) = \sqrt{a}\phi_1(x)$
- ○ $\phi(x) = \phi_1(x) + a$
- ○ $\phi(x) = a^2\phi_1(x)$

(ii) [1 pt] $K(x, z) = aK_1(x, z) + bK_2(x, z)$, for scalars $a, b > 0$

- ○ $\phi(x) = a\phi_1(x) + b\phi_2(x)$
- ○ $\phi(x) = \sqrt{a}\phi_1(x) + \sqrt{b}\phi_2(x)$
- ○ $\phi(x) = a^2\phi_1(x) + b^2\phi_2(x)$
- ○ $\phi(x) = [a\phi_1(x), b\phi_2(x)]$
- ○ $\phi(x) = [\sqrt{a}\phi_1(x), \sqrt{b}\phi_2(x)]$
- ○ $\phi(x) = [a^2\phi_1(x), b^2\phi_2(x)]$

(c) [1 pt] Suppose you are given the choice between using the normal perceptron algorithm, which directly works with $\phi(x)$, and the dual (kernelized) perceptron algorithm, which does not explictly compute $\phi(x)$ but instead works with the kernel function K. Keeping space and time complexities in consideration, when would you prefer using the kernelized perceptron algorithm over the normal perceptron algorithm.

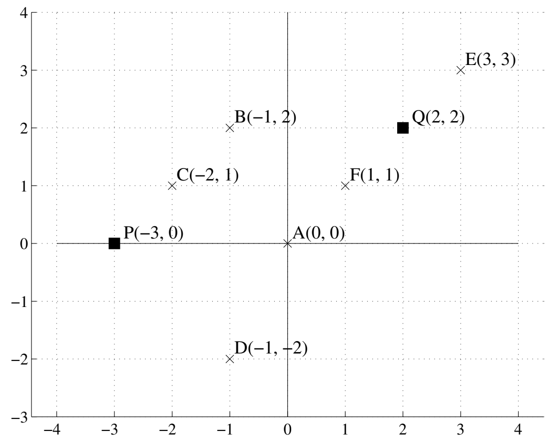
Note: Here N denotes the total number of training samples and d is the dimensionality of $\phi(x)$.

○ $d >> N$ ○ $d << N$ ○ Always ○ Never

Q10. [8 pts] Clustering

In this question, we will do k-means clustering to cluster the points $A, B \ldots F$ (indicated by $\times$'s in the figure on the right) into 2 clusters. The current cluster centers are P and Q (indicated by the ■ in the diagram on the right). Recall that k-means requires a distance function. Given 2 points, $A = (A_1, A_2)$ and $B = (B_1, B_2)$, we use the following distance function $d(A, B)$ that you saw from class,

$$d(A, B) = (A_1 - B_1)^2 + (A_2 - B_2)^2$$



(a) [2 pts] **Update assignment step:** Select all points that get assigned to the cluster with center at P:

○ A ○ B ○ C ○ D ○ E ○ F ○ No point gets assigned to cluster P

(b) [2 pts] **Update cluster center step:** What does cluster center P get updated to?

Changing the distance function: While k-means used Euclidean distance in class, we can extend it to other distance functions, where the assignment and update phases still iteratively minimize the total (non-Euclidian) distance. Here, consider the Manhattan distance:

$$d'(A, B) = |A_1 - B_1| + |A_2 - B_2|$$

We again start from the original locations for P and Q as shown in the figure, and do the update assignment step and the update cluster center step using Manhattan distance as the distance function:

(c) [2 pts] **Update assignment step:** Select all points that get assigned to the cluster with center at P, under this new distance function $d'(A, B)$.

○ A ○ B ○ C ○ D ○ E ○ F ○ No point gets assigned to cluster P

(d) [2 pts] **Update cluster center step:** What does cluster center P get updated to, under this new distance function $d'(A, B)$?

Q5. [6 pts] Kernels and Feature Transforms

A kernel function $K(x, z)$ is a function that conceptually denotes the similarity between two instances x and z in a transformed space. More specifically, for a feature transform $x \to \phi(x)$, the kernel function is $K(x, z) = \phi(x) \cdot \phi(z)$. The beauty of algorithms using kernel functions is that we never actually need to explicitly specify this feature transform $\phi(x)$ but only the values $K(x, z)$ for pairs (x, z). In this problem, we will explore some kernel functions and their feature transforms. For this problem the input vectors are assumed to be 2 dimensional (i.e. $x = (x_1, x_2)$). Remember that $x \cdot z = x_1 z_1 + x_2 z_2$.

(a) For each of the kernel functions below, mark the corresponding feature transform: (mark a single option only for each question)

(i) [1 pt] $K(x, z) = 1 + x \cdot z$

- ○ $\phi(x) = (x_1, x_2)$
- ● $\phi(x) = (1, x_1, x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (x_1^2, x_2^2)$
- ○ $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

(ii) [1 pt] $K(x, z) = (x \cdot z)^2$

- ○ $\phi(x) = (x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- ● $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (1, x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

(iii) [1 pt] $K(x, z) = (1 + x \cdot z)^2$

- ○ $\phi(x) = (1, x_1^2, x_2^2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (1, x_1^2, x_2^2, x_1, x_2, \sqrt{2}x_1 x_2)$
- ● $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (1, x_1, x_2, \sqrt{2}x_1 x_2)$
- ○ $\phi(x) = (1, x_1 x_2, x_1^2 x_2^2)$

For all the above questions, write out $K(x, z)$ and find a $\phi(x)$ such that $K(x, z) = \phi(x) \cdot \phi(z)$. For example in (iii) $K(x, z) = (1 + x_1 z_1 + x_2 z_2)^2 = 1 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2(x_1 z_1 + x_2 z_2 + x_1 x_2 z_1 z_2) = (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2) \cdot (1, z_1^2, z_2^2, \sqrt{2}z_1, \sqrt{2}z_2, \sqrt{2}z_1 z_2)$

(b) Multiple kernels can be combined to produce new kernel functions. For example $K(x, z) = K_1(x, z) + K_2(x, z)$ is a valid kernel function. For the questions below, kernel K_1 has the associated feature transform ϕ_1 and similarly K_2 has the feature transform ϕ_2. Mark the feature transform associated with K for the expressions given below.

Note: The operator $[*, *]$ denotes concatenation of the two arguments. For example, $[x, z] = (x_1, x_2, z_1, z_2)$.

(i) [1 pt] $K(x, z) = aK_1(x, z)$, for some scalar $a > 0$

- ○ $\phi(x) = \phi_1(x)$
- ○ $\phi(x) = [a, \phi_1(x)]$
- ○ $\phi(x) = a\phi_1(x)$
- ● $\phi(x) = \sqrt{a}\phi_1(x)$
- ○ $\phi(x) = \phi_1(x) + a$
- ○ $\phi(x) = a^2\phi_1(x)$

(ii) [1 pt] $K(x, z) = aK_1(x, z) + bK_2(x, z)$, for scalars $a, b > 0$

- ○ $\phi(x) = a\phi_1(x) + b\phi_2(x)$
- ○ $\phi(x) = \sqrt{a}\phi_1(x) + \sqrt{b}\phi_2(x)$
- ○ $\phi(x) = a^2\phi_1(x) + b^2\phi_2(x)$
- ○ $\phi(x) = [a\phi_1(x), b\phi_2(x)]$
- ● $\phi(x) = [\sqrt{a}\phi_1(x), \sqrt{b}\phi_2(x)]$
- ○ $\phi(x) = [a^2\phi_1(x), b^2\phi_2(x)]$

For (ii) we need a ϕ s.t. $\phi(x) \cdot \phi(z) = a\phi_1(x) \cdot \phi_1(z) + b\phi_2(x) \cdot \phi_2(z) = [\sqrt{a}\phi_1(x), \sqrt{b}\phi_2(x)] \cdot [\sqrt{a}\phi_1(z), \sqrt{b}\phi_2(z)]$. Thus we have $\phi(x) = [\sqrt{a}\phi_1(x), \sqrt{b}\phi_2(x)]$

(c) [1 pt] Suppose you are given the choice between using the normal perceptron algorithm, which directly works with $\phi(x)$, and the dual (kernelized) perceptron algorithm, which does not explictly compute $\phi(x)$ but instead works with the kernel function K. Keeping space and time complexities in consideration, when would you prefer using the kernelized perceptron algorithm over the normal perceptron algorithm.

Note: Here N denotes the total number of training samples and d is the dimensionality of $\phi(x)$.

● $d >> N$ ○ $d << N$ ○ Always ○ Never

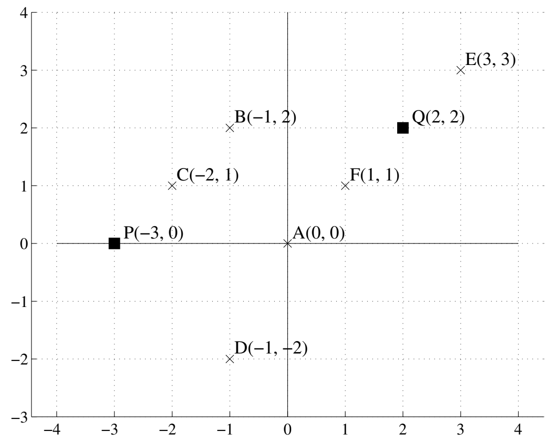
For this question, the rationale was when we use a Kernel function, we typically store a Kernel matrix $\mathbb{K}$ with $\mathbb{K}_{ij} = \phi(x_i) \cdot \phi(x_j)$ where x_i and x_j are the i^{th} and j^{th} training instances. This results in an $N \times N$ matrix. If we were to use the transformed d-dimensional feature representation, we would have to store Nd values instead of N^2 values in the Kernel matrix. Thus space-wise, we would prefer kernels when $d >> N$.

Looking at time complexity, (at test time), if we use kernels (e.g. the kernelized perceptron) we need to compute $\sum_{i=1}^{N} \alpha_{i,y} K(x', x_i)$ for a test sample x'. Assuming the kernel function computation takes $\mathcal{O}(1)$ time, we need to do N such computations. In case of using $\phi(x)$, we have the precomputed weight vector as $w = \sum \alpha_{i,y}\phi(x_i)$ which is d-dimensional and the computation of $w.\phi(x')$ takes d $\mathcal{O}(1)$ computations. So again we would prefer kernels if $d >> N$.

Q10. [8 pts] Clustering

In this question, we will do k-means clustering to cluster the points $A, B \ldots F$ (indicated by $\times$'s in the figure on the right) into 2 clusters. The current cluster centers are P and Q (indicated by the ■ in the diagram on the right). Recall that k-means requires a distance function. Given 2 points, $A = (A_1, A_2)$ and $B = (B_1, B_2)$, we use the following distance function $d(A, B)$ that you saw from class,

$$d(A, B) = (A_1 - B_1)^2 + (A_2 - B_2)^2$$



(a) [2 pts] **Update assignment step:** Select all points that get assigned to the cluster with center at P:

○ A ● B ● C ● D ○ E ○ F ○ No point gets assigned to cluster P

(b) [2 pts] **Update cluster center step:** What does cluster center P get updated to?
The cluster center gets updated to the point, P' which minimizes, $d(P', B) + d(P', C) + d(P', D)$, which in this case turns out to be the centroid of the points, hence the new cluster center is

$$\left(\frac{-1-2-1}{3}, \frac{2+1-2}{3}\right) = \left(\frac{-4}{3}, \frac{+1}{3}\right)$$

Changing the distance function: While k-means used Euclidean distance in class, we can extend it to other distance functions, where the assignment and update phases still iteratively minimize the total (non-Euclidian) distance. Here, consider the Manhattan distance:

$$d'(A, B) = |A_1 - B_1| + |A_2 - B_2|$$

We again start from the original locations for P and Q as shown in the figure, and do the update assignment step and the update cluster center step using Manhattan distance as the distance function:

(c) [2 pts] **Update assignment step:** Select all points that get assigned to the cluster with center at P, under this new distance function $d'(A, B)$.

● A ○ B ● C ● D ○ E ○ F ○ No point gets assigned to cluster P

(d) [2 pts] **Update cluster center step:** What does cluster center P get updated to, under this new distance function $d'(A, B)$?

The cluster center gets updated to the point, P' which minimizes, $d'(P', A) + d'(P', C) + d'(P', D)$, which in this case turns out to be the point with X-coordinate as the median of the X-coordinate of the points in the cluster and the Y-coordinate as the median of the Y-coordinate of the points in the cluster. Hence the new cluster center is

$$(-1, 0)$$