COMP9417 - Machine Learning
Tutorial: Kernel Methods

Weekly Problem Set: Please submit questions 4a and 7 on Moodle by 11:55am Tuesday 29th March, 2022. Please only submit these requested questions and no others.

Question 1 (Dual Perceptron)
Review the development of the Perceptron training algorithm from lectures. Now compare this to the algorithm for Perceptron training *in dual form* introduced in the "Kernel Methods" lecture. The two algorithms are very similar but differ in a few crucial places. Provide an explanation of how the dual version of the algorithm relates to the original.

Question 2 (Feature Transformations)
Recall that the XOR function (graphed below) is not linearly separable. Show how we can learn the XOR function using a linear classifier after applying a feature transformation to the original dataset. As a concrete example, show how to extend the Dual Perceptron from the previous question to learn the XOR function.

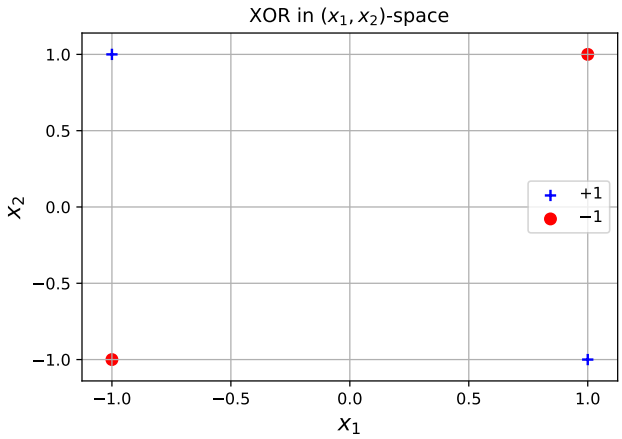


Figure 1: XOR function

Question 3. (The Kernel Trick)
Using the context of the previous question, discuss the computational issues that arise from having to compute high dimensional feature transformations. Show how these can be mitigated by using the Kernel trick, and use this to extend the dual perceptron learning to kernel perceptron learning.

Question 4. (Kernels and their Feature Representations)

In this question, we will show how the choice of kernel gives us different feature transformations. Note that in practice, we will simply choose a kernel and not be too concerned with the exact feature transformation, but it is important to know that different kernels correspond to different representations.

(a) Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ (i.e. $\mathbf{x}$ and $\mathbf{y}$ are two dimensional vectors), and consider the kernel

$$k(\mathbf{x}, \mathbf{y}) = (2\langle \mathbf{x}, \mathbf{y} \rangle + 3)^3.$$

Compute the feature vector $\phi(\mathbf{x})$ correpsonding to this kernel. (In other words, the feature representation of $\mathbf{x}$ and $\mathbf{y}$ such that $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y})$)

(b) **Challenge:** Let $x, y \in \mathbb{R}$, and consider the Gaussian kernel:

$$k(x, y) = \exp\left(-\frac{1}{2\sigma^2}(x - y)^2\right), \qquad \sigma^2 > 0.$$

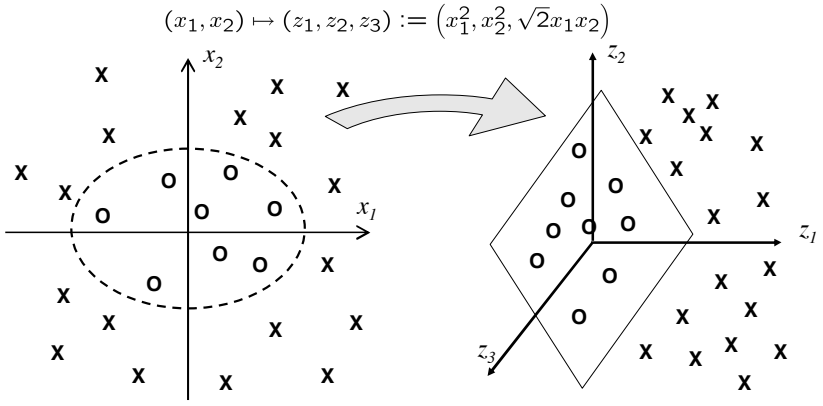
Hint: Use a Taylor expansion to rewrite the exponential in terms of a summation.

Question 5 (More of the Kernel Trick)

You are told that the "kernel trick" means that a non-linear mapping can be realised from the original data representation to a new, implicit feature space simply by defining a kernel function on dot products of pairs of instances from the original data. To see why this is so, you take two instances $\mathbf{x} = [1, 2]^T$ and $\mathbf{y} = [3, 2]^T$, and take their dot product $\langle \mathbf{x}, \mathbf{y} \rangle$ and obtain the answer 7. Clearly, raising this dot product to the power of two will give $(\langle \mathbf{x}, \mathbf{y} \rangle)^2 = 49$. Now expand out this expression to show that this is the same answer you would have obtained if you had simply done a set of feature transformations on the original data.

Question 6 (More Feature Transformations)

Consider the following depiction of a feature transformation from two dimensional space ($\mathbb{R}^2$) to three dimensional space ($\mathbb{R}^3$). Why would we use such a transformation? Generate one example from the $\circ$ class in the original space, and another example from the $\times$ class in the original space, and show their transformed values in the new space.



Question 7 (Support Vector Machines)

The Support Vector Machine is a essentially an approach to learning linear classifiers, but uses a alternative objective function to methods we looked at before, namely *maximising the margin*. Learning algorithms for this problem typically use quadratic optimization solvers, but it is possible to derive the solution manually for a small number of support vectors.

Here is a toy data set of three examples shown as the matrix $\mathbf{X}$, of which the first two are classified as positive and the third as negative, shown as the vector $\mathbf{y}$. Start by constructing the *Gram matrix* for this data, incorporating the class labels, i.e., form the matrix $\mathbf{X}'(\mathbf{X}')^T$. Then solve to find the support vectors, their Lagrange multipliers α, then determine the weight vector $\mathbf{w}$, threshold t and the margin m.

$$\mathbf{X} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix} \qquad \mathbf{X}' = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 0 & -1 \end{bmatrix}$$

To find a maximum margin classifier requires finding a solution for $\mathbf{w}$, t and margin m. For this we can use the following steps (refer to slides 30–35 from the "Kernel Methods" lecture):

1. Set up the Gram matrix for labelled data
2. Set up the expression to be minimised
3. Take partial derivatives
4. Set to zero and solve for each multiplier
5. Solve for $\mathbf{w}$
6. Solve for t
7. Solve for m