CS146 session 6-2 make-up work

Pre-Class:

Parameterize a kernel

1) $k(x,y) = x^T y + c = \phi(x)^T \phi(y)$

   Assume $x = R^2$

   $x^T y + c = x_1 y_1 + x_2 y_2 + c$

   $= (x_1, x_2, c)^T (y_1, y_2, c)$

   $= (x)^T (y)$

   $= \{x_1, x_2, c\}$

2) $k(x,y) = (x^T y)^2 = \phi(x)^T \phi(y)$

   $(x^T y)^2 = (x_1 y_1 + x_2 y_2)^2$

   $= (x_{12} y_{12} + x_{22} y_{22} + 2x_1 x_2 y_1 y_2)$

   $= (x_{12}, x_{22}, 2x_1 x_2)^T (y_{12}, y_{22}, 2y_1 y_2)$

   $= (x)^T (y)$

   $= \{x_{12}, x_{22}, 2x_1 x_2\}$

3) $k(x,y) = \exp(-c\|x-y\|^2) = \phi(x)^T \phi(y)$

   Assuming $x = R$, $c>0$

   $e - c\|x-y\|^2 = e - c(x-y)^2$

   $= e - cx^2 - cy^2 + 2cxy$

   $= (x)^T (y)$

   To expand exponential, use the Taylor series and divide to isolate x and y

   $= e - cx^2\{1, 2cx_1, (2c)^2 2! x_2, (2c)^3 3! x_3, \ldots\}$

Kernel Efficiency

The reason that the kernel is faster is because it does not expend the feature space. Instead, it compute the value of the kernel function. Compare to projecting a kernel which is computational expensive as we need to calculate every point in the projection (if not infinite) as the expansion is non-linear.

Class summery:

Assessment-polls:

1) (same as pre-class), the key is that using the kernel means we are not expending the feature space, and that is the reason that the kernel is much faster.
2) As explained in the video, find the optimum weight is can been down by transferring the prediction vector into the new scale. Which can be calculated by using the inverse of the data matrix multiply the prediction vector, that will give us the optimum weight.

Pre-class Discussion:

For question 1, it is a linear kernel, evaluate the function into vector of x and y. The “c” term is a representation of a constant but more of a representation use. For question three, we should use Taylor expansion.

Decision Function:

The decision boundary is build using sklearn svm function. We time the dot product of x, and the weight. The dual-decision is sum of product the dual-coefficient alpha, and the support vector plus the intercept. The difference between the two is that for the dual-decision function, we are using the sklearn result as the coefficient. It is useful when we have a large feature space, as it is more efficient compare to the first one. For the first one, it is a function with potentially a lot of support vectors, and the second function is with potentially a lot more dimensions. For using the dual-decision function, the phi function is not necessary, as we are in a lower decision space, compare with the fist decision function in the higher dimension (as long as it can be represented as a dot product of two things).

Reflection poll:

In the first activity, we were discussing how to find appropriate parameterizations of kernels mathematically, and in the second activity, by implementing the kernels using decision boundary and dual-decision boundary, we discussed how the different kernel works. By using more support vectors or making the space into a higher dimension.