CS 6362 Machine Learning, Fall 2017: Homework 3

Venkataramana Nagarajan

Question 1:

(a) Given

$$V_s = \frac{r^d \pi^{\frac{d}{2}}}{\tau(\frac{d}{2}+1)}$$

$$V_c = (2r)^d$$

So,

$$\lim_{d\to\infty} \frac{V_s}{V_c} = \lim_{d\to\infty} \frac{\frac{r^d \pi^{\frac{d}{2}}}{\tau(\frac{d}{2}+1)}}{(2r)^d}$$

$$= \lim_{d\to\infty} \frac{\pi^{d/2}}{2^d \tau(\frac{d}{2}+1)} \tag{1}$$

We have been given that

$$\lim_{z\to\infty} \frac{\tau(z+1)}{\sqrt{2\pi z} e^{-z} z^z} = 1$$

which means that for $\lim_{z\to\infty}$

$$\tau(z+1) = \sqrt{2\pi z} e^{-z} z^z$$

So,

$$\tau(\frac{d}{2}+1) = \sqrt{\pi d} e^{-\frac{d}{2}} (d/2)^{\frac{d}{2}} \tag{2}$$

Replacing (2) in (1)

$$\lim_{d\to\infty} \frac{V_s}{V_c} = \frac{\pi^{\frac{d}{2}}}{2^d \sqrt{\pi d} e^{-\frac{d}{2}} (d/2)^{\frac{d}{2}}}$$

Here the term (d/2)$^{\frac{d}{2}}$ in the denominator grows much faster than the terms in the numerator.

Hence,

$$\lim_{d\to\infty} \frac{V_s}{V_c} = 0$$

(b) The relation is that the equation signifies the curse of dimensionality. It shows that as the number of dimensions increase the volume of the sphere becomes insignificant as compared to the volume of the cube. It shows the vast expanse of high-dimensional euclidean space.

Question 2: Here, the cross-validation error is the same for both sets of tuning parameters. So I think if the number support vectors are more it will lead to over-fitting. Also, support vectors are a part of computation when predicting, so more the number of support vectors the more the computation time and cost.

Hence, we should choose the set of parameters which lead to fewer support vectors.

Question 3:

(a) Given

H(a) = max(1-a,0)

So,

H(a) >= 0

Now if 1-a>0 then:

Consider two functions as arguments to a max function, if we prove that the max is a convex then we can say the same for the function.

Consider two generic convex functions f(x) and g(x) as h = max(f(x),g(x))

According to Linear inequality for convexity:

$$h(z) <= th(x) + (1 - t)h(y)$$

where,

$$z = tx + (1 - t)y$$

We can say that

$$f(z) <= th(x) + (1 - t)h(y) \tag{1}$$

$$g(z) <= th(x) + (1 - t)h(y) \tag{2}$$

By convexity of f one knows that:

$$f(z) <= tf(x) + (1-t)f(y)f(z) <= tf(x) + (1-t)f(y) \tag{3}$$

So H(a) is a convex function of a.

(b)

$$\begin{aligned} Max(0.5-a, 0) &= max(1-(a+0.5), 0) \\ &= max(1-a', 0) \end{aligned}$$

(say a' = a+0.5)

$$= H(a')$$

Hence for some value of a' , H' is equivalent to H.

The difference a-a' can be viewed as a difference of λ(a)-λ(a').

Thus we can say that (8) = (9)

Question 4:

(a) Increasing 'd' will make over-fitting more likely. Kernels basically project data into higher dimensional spaces where we may find a linear separator which can be the decision boundary. We use this decision boundary for prediction. But as we increase d, the number of features in the above mentioned higher dimensional space increases substantially. This will make the model to overfit the data.

(b) If we look at the kernel function for gaussian

$$exp(-\frac{||x-x'||^2}{2\sigma^2})$$

This represents the distance between x and x'. If x and x' are very close

$$|x - x'| \approx 0$$

Hence, the gaussian kernel almost equals 1.

But when x and x' are far,

$$|x - x'|$$

is a large number , hence the kernel function is almost zero.

But as σ decreases the kernel function falls even quicker which leads to overfitting of data. So increasing σ would make overfitting less likely.

(c) I assume that the feature maps are finite and non-negative.

Here it is given that:

$$K(x_i, x_i') =< \phi(x_i), \phi(x_i') >$$

So we can say that

$$K_1(x_i, x_i') =< \phi_1(x_i), \phi_1(x_i') >$$

and

$$K_2(x_i, x_i') =< \phi_2(x_i), \phi_2(x_i') >$$

We can write the above in matrix form as follows:

$$K_1(x_i, x_i') + K_2(x_i, x_i') = \begin{bmatrix} \phi_1(x_i) \\ \phi_2(x_i) \end{bmatrix}^T \begin{bmatrix} \phi_1(x_i') \\ \phi_2(x_i') \end{bmatrix}$$

So from above we can say that

$$K_1(x_i, x_i') + K_2(x_i, x_i')$$

is also a kernel function

Question 5:

(a) Linear SVM has a prediction complexity of O(m), where m is the number of features. Since we have n examples the required complexity would be O(mn).

(b) Non linear support vectors have a prediction complexity of O(sm), where s is the number of support vectors and m is the number of features. Since we have n example the required complexity we of O(nsm).