CS 6362 Machine Learning, Fall 2017: Homework 3

Jiani Li

1)

(a) Let $z = d/2$, we got:

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

$$\lim_{d\to\infty} \frac{1}{\Gamma(d/2+1)} = 0$$

And since $\frac{\sqrt{\pi}}{2} < 1$, we get

$$\lim_{d\to\infty} (\frac{\sqrt{\pi}}{2})^d = 0$$

Thus,

$$\lim_{d\to\infty} \frac{V_s}{V_c} = \lim_{d\to\infty} \frac{\frac{r^d \pi^{d/2}}{\Gamma(d/2+1)}}{(2r)^d} = \lim_{d\to\infty} (\frac{\sqrt{\pi}}{2})^d \cdot \frac{1}{\Gamma(d/2+1)} = 0$$

(b) It is much easier to classify if data is concentrate but not sparse. However, as dimensionality increases, the ratio of samples locating inside the d-dimensional sphere will be much smaller than the ratio of samples locating outside the sphere, that is, most data locate far from the center of the feature space, data becomes more and more sparse. Thus, it will be hard to classify those sparse data.

2) We should choose (c_2, γ_2). Because more support vectors means larger margin, meaning we ignore outliers. Therefore, (c_2, γ_2) is less easier to overfitting than (c_1, γ_1).

3)

(a) The definition of convex:

Let X be a convex set in a real vector space and let $f : X \to R$ be a function. f is called convex if:

$$\forall x_1, x_2 \in X, \forall t \in [0,1] : f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

Based on this definition, $h_1(a) = 0$ is convex and $h_2(a) = 1 - a$ is convex. To prove $H(a) = \max(h_1(a), h_2(a))$ is convex, pick any $a_1, a_2 \in R, t \in [0,1]$, then,

$$H(ta_1 + (1-t)a_2) = f_i(ta_1 + (1-t)a_2), \text{for some } i \in \{0,1\}$$

Since h_i is convex, we get

$$\begin{aligned} H(ta_1 + (1-t)a_2) &\leq th_i(a_1) + (1-t)h_i(a_2) \\ &\leq t\max\{h_1(a1), h_2(a1)\} + (1-t)\max\{h_1(a2), h_2(a2)\} \\ &\leq tH(a_1) + (1-t)H(a_2) \end{aligned}$$

Therefore, $H(a)$ is a convex function of a.

(b) Equation (8) is equivalent to

$$\min \lambda\|w\|^2$$
$$\text{subject to } y_i(w^T x_i) \geq 1, (i = 1, \ldots, n)$$

Equation (9) is equivalent to

$$\min \lambda'\|w\|^2$$
$$\text{subject to } y_i(w^T x_i) \geq 0.5, (i = 1, \ldots, n)$$

To make (9) equivalent to (8), $\lambda' = \frac{1}{4}\lambda$

4)

(a) Increasing d makes overfitting more likely. Increasing d will increase the dimensionality of data, which results in overfitting.

(b) Increasing σ makes overfitting less likely. Larger σ means more data can affect the predicting point, whereas smaller σ means only data within a small certain distance from the predicting point can affect the predicting point. Therefore, small σ tends to overfitting.

(c)

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

Let $\phi(x_i) = \begin{bmatrix} \phi_1(x_i) \\ \phi_2(x_i) \end{bmatrix}$, then, $K(x_i, x_i') = \phi(x_i)^T\phi(x_i') =< \phi(x_i)\phi(x_i') >$.

Therefore, $K(x_i, x_i') = K_1(x_i, x_i') + K_2(x_i, x_i')$ is also a kernel function.

5)

(a) The prediction of a linear SVM is $w^T x_i$, which only takes a dot product between weight vector and a test vector. Thus, the computational complexity is $O(m)$.

(b) Compute kernel $K(x_j, x_i)$ takes $O(m)$, thus, making prediction $g(\sum_{j=1}^{n} \alpha_j K(x_j, x_i))$ takes $O(mn)$. But only support vectors' α values are not 0, so if we do not sum up those $\alpha_j = 0$, the time complexity is $O(ms)$.