CS5063: Foundations of Machine Learning

Assignment 2

Submitted By:
Vishal Singh Yadav – CS20MTECH01001

---

1. Let's consider a dataset for which hyperplane $w^T x + b = 0$ exists.

   Now, $w^T x_i + b \geq 1 \; for \; y_i = +1 \; and \; w^T x_i + b \leq -1 \; for \; y_i = -1$ as margins exist.

   $\Rightarrow y_i(w^T x_i + b) \geq 1 \forall i$

   Also $min_{1 \leq i \leq N} |w^T x_i + b| = 1$

   Thus, margin on each side = $min_{1 \leq i \leq N} \frac{|w^T x_i + b|}{||w||} = \frac{1}{||w||}$

   If we change replace 1 in equation of margins by some constant γ then margin equations become,

   $w^T x_i + b \geq \gamma \; for \; y_i = +1 \; and \; w^T x_i + b \leq -\gamma \; for \; y_i = -1$

   $\Rightarrow y_i(w^T x_i + b) \geq \gamma \; \forall i$

   Also $min_{1 \leq i \leq N} |w^T x_i + b| = \gamma$

   Thus, margin on each side = $min_{1 \leq i \leq N} \frac{|w^T x_i + b|}{||w||} = \frac{\gamma}{||w||}$

   We see that the margin on each side of hyperplane changes by factor of γ. Since the margin is changing by a factor on both sides, the line separating the two margins(i.e. a line parallel to both lines and equidistant from both lines) will remain the same. Only the distance of the margins from the hyperplane changes on replacing by a constant.

2. $\rho = \frac{1}{||w||}$

   SVM is the hyperplane which maximises ρ. Maximising $\frac{1}{||w||}$ = minimise $\big|\big|w\big|\big|$ = minimise $\frac{1}{2}\big|\big|w\big|\big|^2$ subject to $\big(w.x_j + b\big)y_j \geq 1, i = 1, \dots, n$

   $$\mathcal{L} = \frac{1}{2}\big|\big|w\big|\big|^2 - \sum_j \alpha_j\big[\big(w^T x_j + b\big)y_j - 1\big] \; where \; \alpha_j \geq 0 \; and \; \alpha_j\big[\big(w^T x_j + b\big)y_j - 1\big] = 0$$

   Put $\nabla_w \mathcal{L} = 0 \; and \; \nabla_b \mathcal{L} = 0$,

   $$w = \sum_j \alpha_j y_j x_j = 0 \tag{1}$$

   $$\sum_j \alpha_j y_j \; = \; 0 \tag{2}$$

$$\Rightarrow \mathcal{L} = \sum_j \alpha_j - \frac{1}{2}\sum_{j,i} \propto_j \propto_i y_j y_i (x_j^T x_i)$$

If (x_j, y_j) are support vectors, then $w^T x^j + b = y_j$

$$b = y_j - \sum_i \alpha_i y_i x_i^T x_j$$

$$\Rightarrow \sum_j \propto_j y_j b = \sum_j \propto_j y_j^2 - \sum_{j,i} \propto_j \propto_i y_j y_i (x_j^T x_i)$$

From (1) and (2)

$$\sum \propto_j - ||w||^2 = 0$$

$$\Rightarrow ||w||^2 = \sum \propto_j$$

Now, $\rho = \frac{1}{||w||}$

$$\Rightarrow \rho^2 = \frac{1}{||w||^2}$$

$$\Rightarrow \frac{1}{\rho^2} = ||w||^2 = \sum_{i=1}^{N} \propto_i$$

3. k_1 and k_2 are valid kernel functions

   a. $k(x, z) = k_1(x, z) + k_2(x, z)$

   Suppose k_1 has feature map α_1 and inner product $<>_{H_{k_1}}$ and $k_2 h$as its feature map α_2 and inner product $<>_{H_{K_2}}$

   Then, $k_1(x, z) =< \alpha_1(x), \alpha_1(z) >_{H_{k_1}}$ and $k_2(x, z) =< \alpha_2(x), \alpha_2(z) >_{H_{k_2}}$

   $\Rightarrow k(x, z) = k_1(x, z) + k_2(x, z)$

   $=< \alpha_1(x), \alpha_1(z) >_{K_{h_1}} + < \alpha_2(x), \alpha_2(z) >_{H_{k_2}}$

   $=< [\alpha_1(x), \alpha_2(x)], [\alpha_1(z), \alpha_2(z)] >_{H_{new}}$

   This means $k(x, z)$ can be expressed as an inner product and hence is valid kernel.

   b. $k(x, z) = k_1(x, z)k_2(x, z)$

   Let α_1 be feature vector of k_1 and α_2 be feature vector of k_2.

   Then, $k_1(x, z) = \alpha_1(x)^T \alpha_1(z)$ and $k_2(x, z) = \alpha_2(x)^T \alpha_2(z)$

   $k(x, z) = k_1(x, z)k_2(x, z)$

$$\Rightarrow \left(\sum_{m=1}^{M} \alpha_{1_m}(x)\alpha_{1_m}(z)\right)\left(\sum_{n=1}^{N} \alpha_{2_n}(x)\alpha_{2_n}(z)\right)$$

$$\Rightarrow \sum_{m=1}^{M}\sum_{n=1}^{N}\left[\alpha_{1_m}(x)\alpha_{2_n}(x)\right]\left[\alpha_{1_m}(z)\alpha_{2_n}(z)\right]$$

$$= \sum_{m=1}^{M}\sum_{n=1}^{N} c_{mn}(x)c_{mn}(z)$$

$c(x)^T c(z)$ where $c_{mn}(i) = \alpha_{1_m}(i)\alpha_{2_n}(i)$

Since we can write $k(x,z)$ as inner product using feature map c, it is a valid kernel.

c. $k(x,z) = h\big(k_1(x,z)\big)$ where h is polynomial function with positive co-efficients

h is given by $a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$ where $a_0 \ldots . a_n > 0$.

$k(x,z) = (a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0)(k\_1(x,z))$ where x^n is product of kernels.

So $k(x,z)$ can be understood as sum of products of kernels with a positive coefficient.

From (b), we know that product of kernels is a valid kernel.

Also from (a), we know that sum of kernels is a valid kernel. Even after adding a positive coefficient, we can still describe sum of kernels as an inner product. Hence sum of kernels is also a valid kernel.

Using the results given above we can conclude that sum of products of kernels is also a valid kernel. Hence we can say $k(x,z) = h\big(k_1(x,z)\big)$ is a valid kernel.

d. $k(x,z) = exp\big(k_1(x,z)\big)$

$$exp(x) = \lim_{i \to \infty}\left(1 + x + \cdots .. + \frac{x^i}{i!}\right)$$

$$exp\big(k_1(x,z)\big) = \lim_{i \to \infty}\left(1 + k_1(x,z) + \cdots .. + \frac{k_1(x,z)^i}{i!}\right)$$

So $k(x,z)$ can be understood as sum of products of kernels with positive coefficient.

From (b), we know that product of kernels is a valid kernel.

Also from (a), we know that sum of kernels is a valid kernel. Even after adding a positive coefficient, we can still describe sum of kernels as an inner product. Hence sum of kernels is also a valid kernel.

Using the results given above we can conclude that sum of products of kernels is also a valid kernel and $k(x,z) = exp\big(k_1(x,z)\big)$ is a valid kernel.

e. $k(x,z) = exp\left(\frac{-||x-z||_2^2}{\sigma^2}\right)$

$$= exp\left(\frac{-||x||_2^2 - ||z||_2^2 + 2x^T z}{\sigma^2}\right)$$

$$= \left(exp\left(\frac{-||x||_2}{\sigma^2}\right) exp\left(\frac{-||z||_2}{\sigma^2}\right)\right) exp\left(\frac{2x^T z}{\sigma^2}\right)$$

From (d) we know that $exp(k_1(x,z))$ is a valid kernel.

From (b), we know that product of kernels is a valid kernel.

Using both these results, we can say that $k(x,z) = exp\left(\frac{-||x-z||_2^2}{\sigma^2}\right)$ is a valid kernel.

4. We have trained the models without shuffling the data.
   a. With linear kernel over entire dataset, we get test accuracy of 0.978774 and the number of support vectors are 28.
   b. In this we have trained on the first *i* points from the training set and tested on all the test points.
      i. For 50 points, we get test accuracy of 0.981132 and the number of support vectors are 2.
      ii. For 100 points, we get test accuracy of 0.981132 and the number of support vectors are 4.
      iii. For 200 points, we get test accuracy of 0.981132 and the number of support vectors are 8.
      iv. For 800 points, we get test accuracy of 0.981132 and the number of support vectors are 14.
   c.
      i. When C = 0.0001, training error @Q=2 = 0.008969 and @Q=5 = 0.004484. Hence, statement is **FALSE**.
      ii. When C = 0.001, support vectors @Q=2 = 76 and @Q=5 = 25. Hence, the statement is **TRUE**.
      iii. When C=0.01, training error @Q=2 = 0.004484 and @Q=5 = 0.003844. Hence, the statement is **FALSE**.
      iv. When C=1, test error @Q=2 = 0.018868 and @Q=5 = 0.021226. Hence, the statement is **FALSE**.

d.

C	Training Error	Test Error
0	0.003844	0.023585
1	0.004484	0.021226
100	0.003203	0.018868
10000	0.002562	0.023585
1000000	0.000641	0.023585

Lowest training error is achieved when C = 1000000.

Lowest test error is achieved when C = 100.

5. Since the test data does not contain labels, we have used validation set for testing.

    a. For linear kernel, out training error is 0.000, test error is 0.024, and the number of support vectors are 1084.

    b. For RBF kernel with gamma set to 0.001, our training error is 0.000, test error is 0.500, and the number of support vectors are 6000.

    For Polynomial kernel with degree = 2 and coef0 = 1, our training error is 0.000, test error is 0.021, and the number of support vectors are 1755.

    We get same training error with both RBF kernel and Polynomial kernel with the only difference being that Polynomial kernel takes less time than RBF kernel.