FoML assignment II : SVM and Kernels

Tamal Mondal

October 6, 2021

1 SVM Theory Question 1

The reason that any scalar like $+\gamma(\gamma > 0)$ or $-\gamma(\gamma > 0)$ can be considered on the right-hand side of the margin boundary equation is, we have the freedom to choose weight matrix w and bias b. As equation $a.x + b.y = c$ is same as $(\frac{a}{c}).x + (\frac{b}{c}).x = 1$, we can always adjust w and b in such a way that right-hand side can be converted to -1 and $+1$.

So for the equation $w.x + b = +\gamma$, we can write $\frac{w}{\gamma}.x + \frac{b}{\gamma} = +1$ or $w^{'}.x + b^{'} = +1$. Similarly for the equation $w.x + b = -\gamma$, we can write $\frac{w}{\gamma}.x + \frac{b}{\gamma} = -1$ or $w^{'}.x + b^{'} = -1$. So the solution for the maximum margin hyperplane remains unchanged even if we consider $+\gamma$ or $-\gamma$ in the right-hand side.

2 SVM Theory Question 2

We know that the following is the primal equation for soft-margin SVM:

$$L_p = \min_{\vec{w},b} \max_{\vec{\alpha} \geq 0} \frac{1}{2} \|w\|^2 - \sum_i \alpha_i [y_i(\vec{w}.\vec{x}_i + b) - 1]$$

Now we know that the SVM hyperplane only depends on support vectors for which $\alpha = 0$ and in that way the 2nd term will be 0(zero) and that would give the optimal solution for L_p as $L_p^* = \frac{1}{2} \|w\|^2$.

Now as we know that the optimization of SVM's maximum margin problem obeys the Slater's condition from convex optimization, so optimal solution for the primal and the dual problem is same($L_p^* = L_d^*$). Given that $\rho =$ half of the margin $= \frac{1}{\|w\|}$.

We can prove the given equation in the question from the dual formulation of the optimization problem and using the KKT conditions as follows,

$$\begin{aligned} L_d &= \max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i.x_j) \text{ such that } \alpha_i \geq 0 \; \forall i \; and \sum_i \alpha_i y_i = 0 \\ &= \max \sum_i \alpha_i - \frac{1}{2} (\sum_i \alpha_i y_i x_i)^2 \\ &= \max \sum_i \alpha_i - \frac{1}{2} \|w\|^2 \text{ (As, one of the KKT condition is } w = \sum_i \alpha_i y_i x_i) \end{aligned}$$

Now as $L_p^* = L_d^*$, we can write,

$$L_d^* = L_p^* = \sum_i \alpha_i - \frac{1}{2}\|w\|^2$$

$$\frac{1}{2}\|w\|^2 = \sum_i \alpha_i - \frac{1}{2}\|w\|^2 \text{ (As, we know } L_p^* = \frac{1}{2}\|w\|^2)$$

$$\|w\|^2 = \sum_i \alpha_i$$

$$\frac{1}{\rho^2} = \sum_i \alpha_i \text{ (As, given that } \rho = \frac{1}{\|w\|})$$

3 Kernel Functions Question

1. For $k(x,z) = k_1(x,z) + k_2(x,z)$, $k(x,z)$ is a **valid** kernel.

   Proof: Let, ϕ_1 and $\langle\rangle_{k1}$ are the feature map and inner product for kernel k_1. Similarly ϕ_2 and $\langle\rangle_{k2}$ are the feature map and inner product for kernel k_2. So, $k_1(x,z) = \langle\phi_1(x), \phi_1(z)\rangle_{k1}$ and $k_2(x,z) = \langle\phi_2(x), \phi_2(z)\rangle_{k2}$. We will follow the same notations for rest of the proofs.

   So, $k(x,z) = \langle\phi_1(x), \phi_1(z)\rangle_{k1} + \langle\phi_2(x), \phi_2(z)\rangle_{k2} = \langle[\phi_1(x), \phi_2(x], [\phi_1(z), \phi_2(z)]\rangle_k = \langle\phi(x), \phi(z)\rangle_k$ which means $k(x,z)$ is also a valid inner product and so a valid kernel.

2. For $k(x,z) = k_1(x,z)k_2(x,z)$, $k(x,z)$ is a **valid** kernel.

   Proof: Multiplying ϕ expressions for kernel k_1 and k_2, we can show that new kernel k is the space of products of the features from ϕ_1 and ϕ_2.

   So, $k(x,z) = \langle\phi_1(x), \phi_1(z)\rangle_{k1}.\langle\phi_2(x), \phi_2(z)\rangle_{k2} = \langle[\phi_1(x), \phi_2(x], [\phi_1(z), \phi_2(z)]\rangle_k = \langle\phi(x), \phi(z)\rangle_k$

3. For $k(x,z) = h(k_1(x,z))$, $k(x,z)$ is a **valid** kernel.

   Proof: Here h is just a transformation in the same domain as kernel k_1 and as every polynomial term is product of kernel having positive coefficients, following the proof of 1 and 2, $k(x,z)$ is also a valid kernel.

4. For $k(x,z) = exp(k_1(x,z))$, $k(x,z)$ is a **valid** kernel.

   Proof: We know $e^x = \lim_{i\to\infty}(1 + x + ... + \frac{x^i}{i!})$. Let's define $H_j(x) = \sum_{i=0}^{j} \frac{x^i}{i!}$ which is nothing but a polynomial with positive coefficients. So as per the previous proof, $H_j(k_1(x,z))$ is also a valid kernel.

   Now, $k(x,z) = exp(k_1(x,z)) = \lim_{j\to\infty} H_j(k_1(x,z))$. So, $k(x,z)$ is also a valid kernel.

5. The given kernel is well known **"Gaussian Kernel"**, which is definitely **valid**.

   Proof: If we simplify the RHS of the kernel function $k(x,z)$, we get,

$$\begin{aligned} k(x,z) &= \exp(\frac{-\|x-z\|_2^2}{\sigma^2}) \\ &= (\exp(\frac{-\|x\|_2^2}{\sigma^2})\exp(\frac{-\|z\|_2^2}{\sigma^2}))\exp(\frac{2x^Tz}{\sigma^2}) \end{aligned}$$

   Now, first two terms together forms a valid kernel as $k_1(x,z) = g(x)g(z)$, for $g : X \to R$, is a valid kernel and from previous proof we know, $exp(k_2(x,z))$ is also a valid kernel(corresponding to 3rd term). Now according to 2, multiplication of two kernels also produce a valid kernel, so $k(x,z)$ is also a valid kernel.

4 SVM Programming Question 1

4.1 Using linear kernel

Using simple linear kernel, I got **97.88%** accuracy during testing and number of support vector was **14** for each of the classes(total **28**).

4.2 Using different size of train data and linear kernel

1. For **training data size = 50**, test accuracy was **98.11%**(same for all 4 scenarios) and **1** support vector for each class(total **2**).
2. For **training data size = 100**, test accuracy was **98.11%** and **2** support vector for each class(total **4**).
3. For **training data size = 200**, test accuracy was **98.11%** and **4** support vector for each class(total **8**).
4. For **training data size = 800**, test accuracy was **98.11%** and **7** support vector for each class(total **14**).

4.3 Using polynomial kernel

1. **False**(training error for Q = 2 and Q = 5 are 0.9% and 0.45% respectively)
2. **True**(number of support vectors for Q = 2 and Q = 5 are [38,38] and [12,13] respectively)
3. **False**(training error for Q = 2 and Q = 5 are 0.45% and 0.38% respectively)
4. **False**(testing error for Q = 2 and Q = 5 are 1.89% and 2.12% respectively)

4.4 Using radial basis function(RBF) kernel

Lowest training error I have got for $C = 10^6$ and it was **0.06%**. Lowest testing error I have got for $C = 100$ and it was **1.89%**.

Here are the training and test errors when $C \in \{0.01, 1, 100, 10^4, 10^6\}$

1. Training and test error for C = 0.01 are **0.38%** and **2.36%** respectively.
2. Training and test error for C = 1 are **0.45%** and **2.12%** respectively.
3. Training and test error for C = 100 are **0.32%** and **1.89%** respectively.
4. Training and test error for C = 10^4 are **0.26%** and **2.36%** respectively.
5. Training and test error for C = 10^6 are **0.06%** and **2.36%** respectively.

5 SVM Programming Question 2

5.1 Using linear kernel

The training error was **0.0%**, test error was **2.4%** and number of support vectors was **542** for each of the class(total **1084**).

5.2 Using Polynomial and Gaussian(RBF) kernel

Using Gaussian(RBF) kernel, the training error was **0.0%**, test error was **50%** and number of support vectors was **3000** for each of the class(total **6000**).

Using Polynomial kernel, the training error was **0.0%**, test error was **2.1%** and number of support vectors was [**817, 938**].

So both the kernels had same **0.0%** training error.