Haiyang Wang
Oct 16th, 2020
hw1927@nyu.edu

Foundations of Machine Learning
Homework 2

Acknowledgements and other stuff

I have discussed problem C with Yucong Lei. Especially, he came up with the notion of $\hat{x}_i$. And this notion helps my computation in C.6.c greatly. Thanks Yucong!

I have attached in the email also my .ipynb file and the zip file of my whole workplace folder for Problem C. In case grader is interested in checking them.

Thanks for grading!

A. Rademacher complexity

1. Nonnegativity of empirical Rademacher complexity

Suppose $S = \{x_1, \cdots, x_m\}$ and $g \in H$, we can easily see that $\forall \sigma$, $\sup_{h\in H} \sum_{i=1}^m \sigma_i h(x_i) \geq \sum_{i=1}^m \sigma_i g(x_i)$.

$$\widehat{\mathfrak{R}}_S(H) = \mathbb{E}_\sigma[\sup_{h\in H} \sum_{i=1}^m \sigma_i h(x_i)] \geq \mathbb{E}_\sigma[\sum_{i=1}^m \sigma_i g(x_i)] = \sum_{i=1}^m g(x_i)\mathbb{E}_\sigma[\sigma_i] = \sum_{i=1}^m g(x_i)\cdot 0 = 0$$

2. Empirical Rademacher complexity for products

Suppose $S = \{x_1, \cdots, x_m\}$. Notice that $\widehat{\mathfrak{R}}_S(H_1) + \widehat{\mathfrak{R}}_S(H_2) = \mathbb{E}_\sigma[\sup_{h\in H_1, g\in H_2} \sum_{i=1}^m \sigma_i(h(x_i) + g(x_i))]$. Further, notice that for 3 possibilities $h(x) + g(x) = 0, 1, 2$:

- if $h(x) + g(x) = 2$, then $h(x)g(x) = 1$
- otherwise, $h(x)g(x) = 0$

We can then define $\Phi(t)$ to be:

- For $t \leq 1$, $\Phi(t) = 0$
- Otherwise, $\Phi(t) = t - 1$

We can easily verify that Φ is a 1-Lipschitz function and $\Phi(h(x) + g(x)) = h(x)g(x)$. Thus we can view $H = \Phi \circ (H_1 \times H_2)$, $h + g \in H_1 \times H_2$, and apply the Talagrand's contraction.

$$\widehat{\mathfrak{R}}_S(H) \leq \widehat{\mathfrak{R}}_S(H_1 \times H_2) = \mathbb{E}_\sigma[\sup_{h\in H_1, g\in H_2} \sum_{i=1}^m \sigma_i(h(x_i) + g(x_i))] = \widehat{\mathfrak{R}}_S(H_1) + \widehat{\mathfrak{R}}_S(H_2)$$

B. VC-dim of neural networks

0. Figure and Terminology

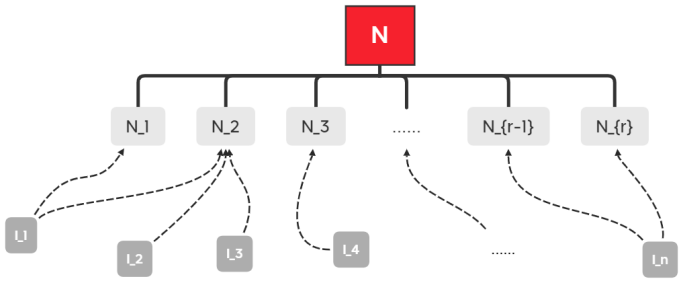

Figure 1: The basic graph of our neural network.

Before solving this problem, I want to firstly clarify some terminology. In Figure 1:

- $I_i, \forall i \in [1, n]$ is the input nodes,
- $N_j, \forall j \in [1, r]$ is the nodes of intermediate layers, each of these has the hypothesis sets H_j. And each of N_j is connected to input nodes $I_{j_1}, \cdots, I_{j_r}$, with $j_1 < j_2 \cdots < j_r \leq n$. Let $\mathbb{R}^r = X_j = \prod_{k=1}^{r} I_{j_k}$. Easily, we can define the function $\pi_j : \mathbb{R}^n \to X_j$, which projects the entire input space to the input subspace that is relevant to N_j. Notice this projection is a surjection.
- N is the output nodes. It has hypothesis set H_0.
- The number of intermediates nodes being r is because that is the exact number of input N can take. Suppose we have more than r intermediates nodes, we can simply ignore those not connected to N. Suppose we have less, then there will be insufficient inputs.
- $H = H' \circ (H_1 \circ \pi_1, H_2 \circ \pi_2, \cdots, H_r \circ \pi_r)$.

1. Upper bounds of growth function with intermediate layers

The growth function of H_j can be written as:

$$\Pi_{H_j}(m) = \max_{\{x_1,\ldots,x_m\} \subseteq X_j} |\{(h(x_1), \ldots, h(x_m)) : h \in H_j\}|$$

For the growth function of H we have :

$$\Pi_H(m) = \max_{\{x_1,\cdots,x_m\} \subseteq X} |\{(h(x_1), \cdots, h(x_m)) | h \in H\}| \tag{1}$$

$$= \max_{\{x_1,\cdots,x_m\} \subseteq X} |\{(h_0 \circ (h_1 \circ \pi_1(x_1), \cdots, h_r \circ \pi_r(x_1)), \cdots, h_0 \circ (h_1 \circ \pi_1(x_m), \cdots, h_r \circ \pi_r(x_m)) | h_j \in H_j, j = 0, \cdots, r\}| \tag{2}$$

$$\leq \max_{\{x_1,\cdots,x_m\} \subseteq X} |\{((h_1 \circ \pi_1(x_1), \cdots, h_r \circ \pi_r(x_1)), \cdots, (h_1 \circ \pi_1(x_m), \cdots, h_r \circ \pi_r(x_m)) | h_j \in H_j, j = 1, \cdots, r\}| \tag{3}$$

$$\leq \prod_{j=1}^{r} \max_{\{x_1,\cdots,x_m\} \subseteq X} |\{(h_j \circ \pi_j(x_1), h_j \circ \pi_j(x_2), \cdots, h_j \circ \pi_j(x_m)) | h_j \in H_j\}| \tag{4}$$

$$= \prod_{j=1}^{r} \max_{\{x_1,\cdots,x_m\} \subseteq X_j} |\{(h_j(x_1), h_j(x_2), \cdots, h_j(x_m)) | h_j \in H_j\}| = \prod_{j=1}^{r} \Pi_{H_j}(m) \tag{5}$$

Explaination

- (1) = (2) is because $H = H' \circ (H_1 \circ \pi_1, H_2 \circ \pi_2, \cdots, H_r \circ \pi_r)$.
- (2) $\leq$ (3) is because H_0 is at most fully shattered. It cannot be more shattered than fully shattered.
- (3) $\leq$ (4) is because $(h_j \circ \pi_j(x_1), h_j \circ \pi_j(x_2), \cdots, h_j \circ \pi_j(x_m))$ is the j^{th} coordinates of $((h_1 \circ \pi_1(x_1), \cdots, h_r \circ \pi_r(x_1)), \cdots, (h_1 \circ \pi_1(x_m), \cdots, h_r \circ \pi_r(x_m))$.
- (4) = (5) is because projection π_j is a surjection.

2. Upper bounds of VCdim of C-neural networks

Because each intermediate level node has VCdim $= d$, by Sauer's lemma, we have:

$$\Pi_{H_j}(m) \leq (\frac{em}{d})^d, \forall m \in \mathbb{N}, \forall j \in [1, \cdots, r]$$

This gives us the bound:

$$\Pi_H(m) \leq \prod_{j=1}^{r} \Pi_{H_j}(m) \leq (\frac{em}{d})^{dr}$$

We now need to find m such that $(\frac{em}{d})^{dr} < 2^m$. Taking $\log_2$ in both side, we can see it is equivalent to find $m > dr \log_2 \frac{em}{d}$.

Using the inequality in the hint, with $x = dr, y = \frac{e}{d}, m = \lceil 2x \log_2(xy) \rceil$. With $r > 1$, we can have our desired bound: VCdim $(H) < m$. In case $r = 1$, the bound is trivially VCdim $(H) < 8d$.

3. Upper bounds of VCdim of H

Theorem 3.13 from the textbook tells that VCdim of hyperplane in $\mathbb{R}^d$ is $d+1$. Thus, our concept class C has VCdim $r + 1$. Applying our previous results by letting $x = (r+1)r, y = e/(r+1), m = \lceil 2x \log_2 xy \rceil$:

$$\text{VCdim } (H) < m = \max(2(r+1)r \log_2(e(r+1)), 8(r+1)) = O(r^2 \log r)$$

C. SVM

I've attached in the email a ipynb file as well as the printed pdf of it, in case grader wants to look at it.

1. Download stuff

Easily Done.

2. Formatting data

Done.

3. Scaling features of all data: use training data for scaling paramter and then apply to test data

Done.

4. Find C and d

The graph I get is:

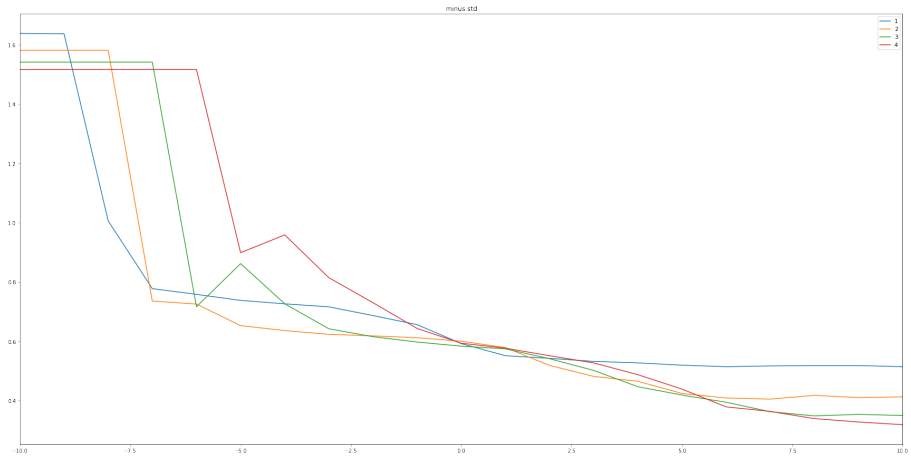

Figure 2: Cross validation error as a function of $\log_2 C$

The range of C is between $2^{-10}, ..., 2^{10}$. And different color of the line represents different value of d. If you look closely you can find the labeling on the up-right direction of the grapha. I choose $(C*, d*) = (2^{10}, 4)$ as my parameters.

5. Plotting tons of graphs

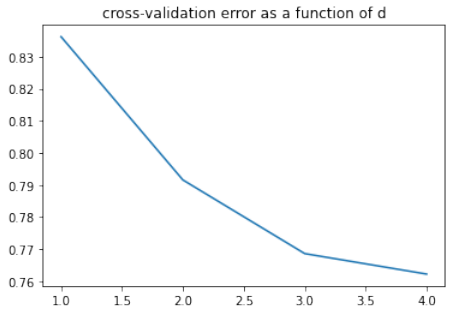

Figure 3: Cross validation error as a function of d

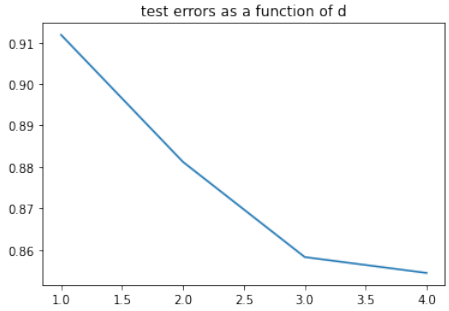

Figure 4: test errors as a function of d

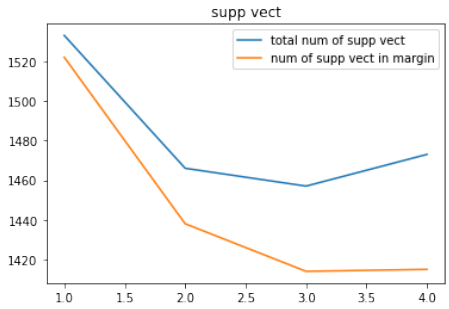

Figure 5: number of unbounded or bounded support vectors as a function of d

6. Minimizing sparsity

The sparsity maximizing, or the $|\alpha|_2$ minimizing optimization problem is the following:

$$\begin{gathered}\min_{\alpha,b,\xi} \frac{1}{2}\sum_{i=1}^{m}\alpha_i^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to } y_i\left(\sum_{j=1}^{m}\alpha_j y_j K(x_i,x_j)+b\right) \geq 1-\xi_i, i\in[m] \\ \xi_i,\alpha_i \geq 0, i\in[m]\end{gathered} \tag{6}$$

The primal optimization problem of SVM is:

$$\begin{gathered}\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i^p \\ \text{subject to } y_i\left(\sum_{j=1}^{m}\alpha_j y_j K(x_i,x_j)+b\right) \geq 1-\xi_i \wedge \xi_i \geq 0, i\in[m]\end{gathered} \tag{7}$$

(a) Coincides with primal optimizaion?

Problem (6) modulo the condition $\alpha_i \geq 0$ becomes:

$$\begin{gathered}\min_{\alpha,b,\xi} \frac{1}{2}\sum_{i=1}^{m}\alpha_i^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to } y_i\left(\sum_{j=1}^{m}\alpha_j y_j K(x_i,x_j)+b\right) \geq 1-\xi_i \wedge \xi_i \geq 0, i\in[m]\end{gathered} \tag{8}$$

For problem (7): We have the formula $\mathbf{w} = \sum \alpha_i y_i \mathbf{x_i}$. With the assumption that $\mathbf{x_i}$ forms an ortho-normal basis of $\mathbb{R}^m$, we can know $|w|^2 = \sum |\alpha_i y_i|^2 = \sum \alpha_i^2$. Further, lets $p = 1$. The problem (7) can be seen in the exact same form of (8). This tells us that (8) is a special case of (7).

(b) PSD or not?

The requirements of a convex optimization problem are: 1. the objective function must be convex. 2. the domain defined by the constraints must be convex. Kernel being positive-definite or not does not effect the convexity of objective function. So we only need to check that if the PSD condition effect the convexity of the domain.

Suppose we have 2 points satisfying the constrains: $A = (\alpha_1, \cdots, \alpha_m, b, \xi_1, \cdots, \xi_m)$ and $B = (\alpha'_1, \cdots, \alpha'_m, b', \xi'_1, \cdots, \xi'_m)$. For any $\lambda \in (0,1)$. We need to check if $\lambda A + \lambda B$ satisfies the constraints. This is equivalent of varifying that

$$y_i(\sum_{j=1}^{m}(\lambda\alpha_j + (1-\lambda)\alpha'_j)y_j K(x_i,x_j) + b) \geq 1 - (\lambda\xi_i + (1-\lambda)\xi'_i), i\in[m] \tag{9}$$

$$\lambda\xi_i + (1-\lambda)\xi'_i \geq 0, i\in[m] \tag{10}$$

The (11) is clearly correct, now I will focus on (10). For any $i \in [m]$, the LHS of (10) is:

$$\lambda y_i\left(\sum_{j=1}^{m}\alpha_j y_j K(x_i,x_j)+b\right) + (1-\lambda)y_i\left(\sum_{j=1}^{m}\alpha'_j y_j K(x_i,x_j)+b\right)$$

By the fact that A, B satisfy the constraints, we can know that:

- $y_i\left(\sum_{j=1}^{m}\alpha_j y_j K(x_i,x_j)+b\right) \geq (1-\xi_i)$
- $y_i\left(\sum_{j=1}^{m}\alpha'_j y_j K(x_i,x_j)+b\right) \geq (1-\xi'_i)$

Thus the LHS of (10) is greater than:

$$\lambda(1-\xi_i) + (1-\lambda)(1-\xi'_i) = 1 - (\lambda\xi_i + (1-\lambda)\xi'_i)$$

which is the RHS of (10). Thus we have shown that (10) is true.

From the above proof we can see the PSDness of K does not related to the convexity of the domain. So positive-definiteness of K is not needed to ensure this is a convex optimization problem.

(c) The dual optimization of (6)

There are three groups of constraints:

- $f_i(\alpha, b, \xi) = y_i \left(\sum_{j=1}^{m} \alpha_j y_j K(x_i, x_j) + b \right) - 1 + \xi_i \geq 0$. I will assign it with the parameter A_i in Lagrangian.
- $\alpha_i \geq 0$. I will assign it with the parameter B_i in Lagrangian.
- $\xi_i \geq 0$. I will assign it with the parameter D_i in Lagrangian.

Thus, we can have the Lagrangian:

$$L(\alpha, b, \xi, A, B, D) = \frac{1}{2}||\alpha||^2 + \sum_{i=1}^{m}(C\xi_i - A_i f_i(\alpha, b, \xi) - B_i\alpha_i - D_i\xi_i)$$

Taking $\nabla_\alpha, \nabla_b, \nabla_\xi$ to L gives us following condition:

- $\alpha = B + \sum_{i,j=1}^{m} A_i y_i y_j K(x_i, x_j) \cdot x_j$
- $\sum_{i=1}^{m} A_i y_i = 0$, which is basiclly saying A is perpendicular to y.
- $C = A_i + D_i, \forall i \in [m]$

Before trying to do the calculation, I want to define some short-hands:

- $K(x_i, x_j) = K_{i,j}$
- $\hat{x}_j = (y_1 K_{j,1}, y_2 K_{j,2}, \cdots, y_m K_{j,m})$
- $z = (A_1 y_1, A_2 y_2, \cdots, A_m y_M)$
- $\hat{x}$ is the matrix defined as $\hat{x}_{i,j} = \hat{x}_{ij}$.

For example, I can write: $\alpha_i = B_i + z \cdot \hat{x}_i$ now

Plugging $\alpha = B + \sum_{i=1}^{m} z \cdot \hat{x}_i \cdot x_i$ into the expression of L we can have :

$$L = \frac{1}{2}\sum_{i=1}^{m}(B_i + z\cdot\hat{x}_i)^2 + \sum_{i=1}^{m} C\xi_i - \sum_{i=1}^{m} B_i(B_i + z\cdot\hat{x}_i) - \sum_{i=1}^{m} D_i\xi_i - \sum_{i=1}^{m} A_i \left(y_i \left(((\sum_{j=1}^{m}(B_j + z\cdot\hat{x}_j)y_j K_{i,j}) + b) - 1 + \xi_i \right) \right)$$

I can simplify this equation as:

$$L = \frac{1}{2}\sum_{i=1}^{m}(B_i + z\cdot\hat{x}_i)^2 - \sum_{i=1}^{m}B_i(B_i + z\cdot\hat{x}_i) - \sum_{i=1}^{m}A_i\left(y_i\left((\sum_{j=1}^{m}(B_j + z\cdot\hat{x}_j)y_jK_{i,j}) + b\right) - 1\right)\right) \tag{11}$$

$$= \frac{1}{2}\sum_{i=1}^{m}(B_i + z\cdot\hat{x}_i)^2 - \sum_{i=1}^{m}B_i(B_i + z\cdot\hat{x}_i) + \sum_{i=1}^{m}A_i - \sum_{i=1}^{m}A_i\left(y_i\left((\sum_{j=1}^{m}(B_j + z\cdot\hat{x}_j)y_jK_{i,j}) + b\right)\right) \tag{12}$$

$$= \frac{1}{2}\sum_{i=1}^{m}(B_i + z\cdot\hat{x}_i)^2 - \sum_{i=1}^{m}B_i(B_i + z\cdot\hat{x}_i) + \sum_{i=1}^{m}A_i - \sum_{i,j=1}^{m}A_iy_iy_j(B_j + z\cdot\hat{x}_j)K_{i,j} \tag{13}$$

$$= -\frac{1}{2}\sum_{i=1}^{m}B_i^2 + \frac{1}{2}\sum_{i=1}^{m}(z\cdot\hat{x}_i)^2 + \sum_{i=1}^{m}A_i - \sum_{i,j=1}^{m}A_iy_iy_j(B_j + z\cdot\hat{x}_j)K_{i,j} \tag{14}$$

$$= \frac{1}{2}\sum_{i=1}^{m}(B_i + z\cdot\hat{x}_i)^2 - \sum_{i=1}^{m}B_i(B_i + z\cdot\hat{x}_i) + \sum_{i=1}^{m}A_i - \sum_{i,j=1}^{m}A_iy_iy_j(B_j + z\cdot\hat{x}_j)K_{i,j} \tag{15}$$

$$= -\frac{1}{2}\sum_{i=1}^{m}B_i^2 + \frac{1}{2}\sum_{i=1}^{m}(z\cdot\hat{x}_i)^2 + \sum_{i=1}^{m}A_i - \sum_{i,j=1}^{m}A_iy_iy_j(B_j + z\cdot\hat{x}_j)K_{i,j} \tag{16}$$

$$= \sum_{i=1}^{m}A_i - \frac{1}{2}\sum_{i=1}^{m}B_i^2 - \frac{1}{2}\sum_{i=1}^{m}(z\cdot\hat{x}_i)^2 - \sum_{i\neq j}^{m}A_iy_iy_j(B_j + z\cdot\hat{x}_j)K_{i,j} \tag{17}$$

$$= \sum_{i=1}^{m}A_i - \frac{1}{2}||B + z\cdot\hat{x}||^2 \tag{18}$$

Thus the dual problem is

$$\max_{A,B}\sum_{i=1}^{m}A_i - \frac{1}{2}||B + z\cdot\hat{x}||^2$$

Under the constraints:$B_i \geq 0, 0 \leq A_i \leq C, A\cdot y = 0$

(d) Omitting non-negativity constraint on α. Use libsvm to solve the problem. Plot ten-fold cross-validation training and test errors for the hypotheses obtained based on the solution α as a function of d, for the best value of C measured on the validation set

This problem requires me to modify the code in libsvm/svm.cpp. However, I am not so familiar cpp, and unable to modify the code.