Solutions for Homework 2

Bao Jinge A0214306U e0522065@u.nus.edu

1 MDL with multiplicative error bound

1.1 a

Using Haussler's inequality, let $L_D(h) = \mu$ and $L_S(h) = \hat{\mu}$, we have

$$Pr(|L_S(h) - L_D(h)| \geq \alpha(\epsilon + \mu + \hat{\mu})) \leq 2e^{-\frac{\alpha^2 \epsilon m}{M}}$$

Let RHS is equal to δ, we get sample complexity

$$m_{\mathcal{H}}(\epsilon, \delta) = \frac{M}{\alpha^2 \epsilon} \ln \frac{2}{\delta}$$

So, for any $h \in \mathcal{H}$, if sample $m > m_{\mathcal{H}}(\epsilon, \delta)$, with probability at least $1 - \delta$, we could have

$$L_{\mathcal{D}}(h) \leq \frac{1+\alpha}{1-\alpha} L_{\mathcal{S}}(h) + \frac{\alpha}{1-\alpha}\epsilon$$

Considering using SRM, let $\mathcal{H} = \cup_{i=1}^{n} \mathcal{H}_n$, where each $\mathcal{H}_n$ is a singlton and $\mathcal{H}$ is conuntable. By SRM rule, we need to upper bound all hypothesises class in each $\mathcal{H}_n$. Let $\sum_{i=1}^{n} w(n) \leq 1$ and use union bound, then we know $\forall h \in \mathcal{H}_n$

$$L_{\mathcal{D}}(h) \leq \frac{1+\alpha}{1-\alpha} L_{\mathcal{S}}(h) + \frac{\alpha}{1-\alpha} \min_{n: h \in \mathcal{H}_n} \epsilon_n(m, w(n)\delta)$$

As analysis in Chapter 7.3 from SSBD, since each singleton should have the property of uniform convergence, we let

$$\min_{n: h \in \mathcal{H}_n} \epsilon_n(m, w(n)\delta) = \frac{M}{\alpha^2 m} \ln(\frac{2}{w(n)\delta})$$

Using MDL rule and each hypothesis class is a singleton, we set $w(n) = w(h) = \frac{1}{2^{|h|}}$, so the new upper bound is as follows

$$L_{\mathcal{D}}(h) \leq \frac{1+\alpha}{1-\alpha} L_{\mathcal{S}}(h) + \frac{M(\ln(2/\delta) + |h|)}{1-\alpha}$$

1.2 b

From what section (a) tell us, the estimation error in such upper bound is

$$\epsilon_{est} = \frac{2\alpha}{1-\alpha} L_{\mathcal{S}}(h) + \frac{M(\ln(2/\delta) + |h|)}{1-\alpha}$$

Thus, we can construct regularized cost function as follows

$$l(h) = L_{\mathcal{S}}(h) + \lambda|h|$$

where λ is a constant coefficient.

1.3 c

In the bound derived from Hoeffding's Inequality, the estimation error grows in $\mathcal{O}(\frac{1}{\sqrt{m}})$. However, in the new bound rederived from Haussler's Inequality, the esimation error grows in $\mathcal{O}(\frac{1}{m})$, but with approximation error rescaled by $\frac{1+\alpha}{1-\alpha} > 1$.
From what has been discussed above, we may safely draw a conclusion.
When $L_{\mathcal{S}}(h)$ is small or zero, then the upper bound basically depends on estimation error as m increase. Thus, the new upper bound rederived (from Haussler's Inequality) will fit well.
When $L_{\mathcal{S}}(h)$ is large, the approximation error will contribute more to the upper bound then estimation error, with scale $\frac{1+\alpha}{1-\alpha} > 1$, the former bound (derived from Hoeffding's Inequality) will be better.

2 Margin VS Network Size

2.1 a

From definition of VC-Dimention, we just need find a example of k instances that our convex polygon classifier can shatter. Thus, we constrcut a circle on the plane, and let all k instances located on the circle. We can prove that, no matter how every instance is labeled positive or negative, we can shatter this instance set by connectting the all positive point clockwisely or counter-clockwisely, which is formed into a convex polygon. Obviously, all negative points will be outside this convex polygon and all positive points will be on the boundary of this convex polygon. Futhermore, if we have all k instances labelled by positive, than our built convex polygon classifier will have k vertices. If we have less than k instances labelled by positive, then convex polygon built by this way will have vectices less than k. If we have no positive instances, then convex polygons of any vertices will shatter this set (just let convex polygon inside the circle). To sum up, According to the difition of VC dimension, the VC demension of convex poly with k or fewer vertices is at least k.

2.2 b

Because every linear threhold function could partition the space into two half-space, with vector of instance (i.e. point in a plane in this senario) $p = (x_1, x_2)^T$ inserted into input layer (just one input units), we could construct k connects between each linear hidden units and input units with specified weights and design an activation function $\sigma_i(p)$ for each linear threshold hidden units $i \in [k]$ that

$$\sigma_i(p) = \begin{cases} 1, & \langle w^i, p \rangle \geq b^i \\ 0, & otherwise \end{cases}$$

where bias b^i and weight vector w^i correspond to each connections between input unit and corresponding hidden unit. Obviously , each connection detones a linear threshold function. Then we connect each hidden units to output layar which has only one output unit with all weights set to 1(like a AND function). Let h denotes the vector outputs of each hidden unit. Then we output the label

$$g_{out}(h) = \begin{cases} 1, & \sum_{i=0}^{k-1} \geq k \\ -1, & otherwise \end{cases}$$

where label 1 denotes positive and -1 denotes negative. As we can see, $g_{out}(h)$ is also a linear threshold funciton, where coordinates of its weight vector in this function is also 1, and bias is k. To clarify, because every hidden linear threshold unit can partition space into two halfspaces, all k linear threshold functions construct a convex ploygon with at most k vetices. And the point through hidden layer with all outputs 1 will be inside the convex polygen (including boundary) undoubtedly. The condition $\sum_{i=0}^{k-1} \geq k$ in function $g(h)$ means the points inside the convex polygon represented as the intersection of k halfplane is positive (including on the boundary). Otherwise, the points will be negative, which mean, the point is outside this convex polygon. With the result of (a), we can safely say that the VC-dimension of a single hidden layer neural networks with k linear threshold hidden units is at least k.

2.3 c

From definition of VC-dimension, the VC-dimension of linear threhold function on 2-dimensional vector is 3. As what Sauer's Lemma tells us, the upper bound of the growth function is

$$\tau_{\mathcal{H}}(m) \leq (\frac{em}{3})^3$$

which indicates upper bound of the number of hidden units that each determine a different function is no more than $(em/3)^3$. Namely, the output vector of hidden units have dimension no more then $(em/3)^3$. Suppose h denotes the output vector of hidden units and this single hidden layer neural network is $\mathcal{H}$. According to the Lemma 26.11 from SSBD's book, suppose this single hidden layer neural network to $\mathcal{H}$. If we have sample set $S = (p_1, p_2, ..., p_m)$, then output of hidden layer will have $S_h = (h_1, h_2, ..., h_m)^T$ and each example is a vectors in $\{0, 1\}^d$ (to fit the theorem, we bound by $\mathbb{R}^d$), where d is $(em/3)^3$, because of output weights of hidden unit have l_1 norm 1, so the $\max_i \|h_i\|_\infty = 1$. Then Rademacher Complexity of this single hidden layer neural networks $\mathcal{H}$ are bounded as follows

$$\mathcal{R}(\mathcal{H} \circ \mathcal{S}) \leq \sqrt{\frac{2\log 2(\frac{em}{3})^3}{m}} = \sqrt{\frac{2\log 2 + 6\log(em/3)}{m}}$$

2.4 d

According to the Theorem 26.15 from SSBD's book, we can get a bound on the error. Using Hinge loss here, we have Lipschitz constant $\rho = 1$. With single hidden layer has l_1 norm equal to 1, we have $R = 1$ satisfying $R \geq \|h_i\|_\infty$. Suppose the margin of neural networks is $B = \frac{1}{\gamma}$, $c = 1 + (1/\gamma)$ and $d = (em/3)^3$. For $h \in \mathcal{H}$, which is such single hidden layer neural networks we have bound as follow

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + \frac{2}{\gamma}\sqrt{\frac{2\log 2 + 6\log(em/3)}{m}} + (1 + \frac{1}{\gamma})\sqrt{\frac{2\ln(2/\delta)}{m}}$$

2.5 e

According to Chapter 28.1 from SSBD, using Lemma 26.8 and Theorem 26.5, we obtain that with probability of $1-\delta$, for every $h \in \mathcal{H}$, we have that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + \sqrt{\frac{8D\log(em/D)}{m}} + \sqrt{\frac{2\log 2/\delta}{m}}$$

Here D denotes VC-dimension.
To sum up, if we know that the margin will be larger (which means instances are linearly separated w.h.p.) for the instacnes and the VC-dimension of the single hidden layer neural networks will be large, then the former bound will be better. However, if we know that the margin will be smaller (which means instances can not be linearly separated well) for the instacnes and the VC-dimension of the single hidden layer neural networks will be small, then the latter bound will be better. Otherwise, the two upper bound will be both good.

3 Estimation Error for l1 vs l2 regularization

3.1 a

Suppose there are k d-length strings. Suppose w is a $k+1$-demensional weight vector of SVM where w_0 is the bias. Let is kernel function which maps the x to a featture vector

$$\psi(x) = (\psi_{u0}(x), \psi_{u1}(x), \psi_{u2}(x), ..., \psi_{uk}(x))^T$$

which is a $(k+1)$ dimensional vector, where the first coordinate $\psi_{u0}(x)$ is a constant function with value 1 (corresponding to the bias w_0). The target function of this hard SVM is

$$f_v(x) = sign(< w, \psi(x) > +b)$$

Because of magnitute is at least 1, so we let w_v is 2, bias w_0 is -1 and other coordinates are 0. According to the Theorem 26.12 from SSBD, we have a bound for l_2 regularization such that $\forall w' \in \mathcal{H}$,

$$L_{\mathcal{D}}(w') \leq L_{\mathcal{S}}(w') + \frac{2\rho BR}{\sqrt{m}} + c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

Since this is a hard SVM model, the value$f_v(x)$ is either 0 or 1. Because each coordinate of $\psi(x)$ is a indicate value (either 0 or 1) and $\|\psi(x)\|_2 \leq \sqrt{F}$, we set $R = \sqrt{F}$. As $\|w\|_2 = \sqrt{5}$, we set $R = \sqrt{5}$. We use the Hinge loss to upper bound the 0-1 loss. As Hinge loss has is 1-Lipschitz, so $\rho = 1$ and $c = 1 + \sqrt{5F}$. Consequently, we have more accurate upper bound that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + 2\sqrt{\frac{5F}{m}} + (1+\sqrt{5F})\sqrt{\frac{2\ln(2/\delta)}{m}}$$

3.2 b

In this case, the target function which we want to learn is the same as target function $f_v(x)$ in section (a). According to Theorem 26.15 from SSBD, we have a bound for l_1 regularization such that $\forall w' \in \mathcal{H}$,

$$L_{\mathcal{D}}(w') \leq L_{\mathcal{S}}(w') + 2\rho BR\sqrt{\frac{2\log(2d')}{m}} + c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

Because $\|\psi(x)\|_\infty = 1$ and $\|w\|_1 \leq 3$, we set $R = 1$ and $B = 3$. Using Hinge loss to upper bound the 0-1 loss, we have $\rho = 1$ and $c = 4$. As the number of possible characters is C, there are at most C^d words of d length. In other way, the dimensional of $\psi(x)$ is at most $C^d + 1$ (including the first coordinate corresponding to bias w_0). So we have upper bound such that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + 6\sqrt{\frac{2\log 2 + 2d\log C}{m}} + 4\sqrt{\frac{2ln(2/\delta)}{m}}$$

Comparing the above bound to bound in (a), we can find that the former bound is mainly determined by length of file F, but the latter bound is mainly determined by the number of character C and length d. Because many words will come out duplicately in a same file, we could $C^d \ll F$. Thus, the latter upper bound will be tighter than the former in section (a).

3.3 c

In this case, because there are $|S|$ virus string, the target function will have $|S|$ coordinates equal to 2. Consequently, $\|w\|_2 = \sqrt{1 + 4|S|}$, but $\|w\|_1 = 1 + 2|S|$. Correspondingly, by using Hinge loss, we have $\rho = 1$. For l_2 regulartization, $B = \sqrt{1 + 4|S|}$,$R = \sqrt{F}$, $c = 1 + \sqrt{(1 + 4|S|)F}$. For l_1 regularization we have $B = 1 + 2|S|$, $R = 1$, $c = 2 + 2|S|$.
For l_2 regularization, we have upper bound as follows

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + 2\sqrt{\frac{(1 + 4|S|)F}{m}} + (1 + \sqrt{(1 + 4|S|)F})\sqrt{\frac{2\ln(2/\delta)}{m}}$$

For l_1 regularization, we have upper bound as follows

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{S}}(h) + 2(1 + 2|S|)\sqrt{\frac{2\ln 2 + 2d\ln C}{m}} + (2 + 2|S|)\sqrt{\frac{2ln(2/\delta)}{m}}$$

In first bound, the estimation error is in $O(\sqrt{|S|})$. But the estimation error is in $O(|S|)$ in second bound. When comparing these two bound, because $|S|$ is much larger than F, the former upper bound for l_2 regularization will be better.