Assignment 3, Machine Learning Fall 2018

Cristian Mitroi, dmn470

11 December 2018

Contents

1

We select $S = \{0, 1\}$ with $\mathbb{E}[X] = 0.5$, and we apply the following rule to the sampling algorithm:

- we first sample randomly from that space, either a 0 or a 1;
- all the other samples depend on this initial one:
  - if it's 0 then they are all 0
  - if it's 1 then they are all 1

In this case we get that $\frac{1}{n}\sum_{i=1}^{n} X_i =$

- 0, if they are all 0
- 1, if they are all 1

Then $|\mu - \frac{1}{n}\sum_{i=1}^{n} X_i| \geq 0.5$

See `ex1.py` for simulations.

2 To Split or Not To Split?

2.1

We know that H is a finite set, $|H| = M$. Thus we can use the "Generalization Bound for Finite Hypothesis Classes" (**Theorem 3.2.**). We have:

$$P(L(h) \leq \hat{L}(h, S) + \sqrt{\frac{\ln M/\delta}{2n}}) \geq 1 - \delta \tag{1}$$

2.2

Since each of the hypotheses are trained on a separate set, we use "Generalization Bound for a Single Hypothesis" (**Theorem 3.1.**). However, the sample size is smaller, so instead of $2n$ we have $2n/M$. This results in:

$$P(L(h) \leq \hat{L}(h, S) + \sqrt{\frac{\ln 1/\delta}{2n/M}}) \geq 1 - \delta \tag{2}$$

We can see that the bound is looser in this case, since $2n/M$ in the denominator increases the $\sqrt{}$ confidence interval. In this sense, the proposal is not really a good idea.

2.3

2.3.a

First we select one hypothesis out of a set of multiple hypotheses by validating on the same sample set. Thus we are using **Theorem 3.2.** again. In our case the sample size $= 2n/2 = n$, since we split the validation set in 2 equal subsets. So:

$$P(L(h) \leq \hat{L}(h, S^1_{val}) + \sqrt{\frac{\ln(M/\delta)}{n}}) \geq 1 - \delta \tag{3}$$

For the second bound we are using **Theorem 3.1.**, since it's only one hypothesis:

$$P(L(h) \leq \hat{L}(h, S^2_{val}) + \sqrt{\frac{\ln(1/\delta)}{n}}) \geq 1 - \delta \tag{4}$$

2.3.b

In order for the bound in eq. 1 to be tighter than eq. 4 we need to have:

$$\sqrt{\frac{ln\frac{M}{\delta}}{2n}} < \sqrt{\frac{ln\frac{1}{\delta}}{n}}$$

We can derive M as a function of δ analytically:

$$\implies \frac{ln\frac{M}{\delta}}{2n} < \frac{ln\frac{1}{\delta}}{n} \implies ln\frac{M}{\delta} < 2ln\frac{1}{\delta}$$

$$\implies ln\frac{M}{\delta} < ln(\frac{1}{\delta})^2 \implies \frac{M}{\delta} < \frac{1}{\delta}^2 \implies M < \frac{1}{\delta}$$

Thus for any $M < \frac{1}{\delta}$ the bound in eq. 1 is tighter. For any value of $M > \frac{1}{\delta}$ the value in eq. 4 will be tighter. They are equal when $M = \frac{1}{\delta}$.

2.3.c

First of all, as the question states, we are interpreting this question as not being about comparing eq. 1 and eq. 4 anymore. It's about analyzing the procedure in *3* for any *other* drawbacks.

These could be:

Firstly, the validation set size is smaller. This yields a looser bound for the probability.

Secondly, we can think of S^2_{val} as a test set. This is because it's not used in selecting a hypothesis, but as a final measure of the quality of our selected hypothesis and as an approximation of L (expected loss). So, we are relying too much on the bound placed on this (the one in eq. 4). However, it is important to consider the hypothesis selection process, which is expressed in eq. 3 . Simply reporting eq. 4 does not capture this process.

2.4

2.4.a

If we choose an α close to 1 for the S^1_{val} then we end up with a more robust hypothesis (since it's been validated on more data and its bound will be tighter). However, S^2_{val} (the "test" set) will not provide any meaningful metric on its bound. It will have only a few samples and thus will be very loose.

The opposite would be the case if we chose an α close to 0 for S^1_{val}. The model would be selected on a very loose bound, thus most likely rendering it useless. However, this choice would yield a larger "test" set, so the bound on this will be more meaningful (tighter). This will perhaps in turn highlight how the hypothesis is weak, alarming the user to his poor decision process.

2.4.b

Personally, I would prefer a more reliable hypothesis to begin with. Thus I would choose an α close to 1. However, I do consider the "test" metric (the bound on S^2_{val}) to be valuable in expressing a final confidence about your algorithm. In the end, I would prefer to have an equally tight bound on eq. 3 and eq. 4 . In this way they are both equally reliable.

Thus I would extract α based on the equality between bounds eq. 3 and eq. 4 . Observe that α is actually applied to the full sample size of $2n$:

$$\sqrt{\frac{ln\frac{M}{\delta}}{\alpha 2n}} = \sqrt{\frac{ln\frac{1}{\delta}}{(\alpha-1)2n}} \implies (\alpha-1)ln\frac{M}{\delta} = \alpha ln(\frac{1}{\delta}) \implies (\frac{M}{\delta}^{\alpha-1}) = (\frac{1}{\delta})^{\alpha}$$

$$\implies \frac{M^{\alpha-1}}{\delta^{\alpha-1}} = \frac{1}{\delta^{\alpha}} \implies \delta^{\alpha}M^{\alpha-1} = \delta^{\alpha-1} \implies M^{\alpha-1} = \delta^{\alpha-1}/\delta^{\alpha}$$

$$\implies M^{\alpha-1} = \delta^{-1} \implies \alpha - 1 = log_M\frac{1}{\delta} \implies \alpha = log_M\frac{1}{\delta} + 1$$

3 Occam's Razor

3.1

Inspired by the binary tree application in the lecture notes, we see that $|H_d| = 2^{27^d}$. We define $\pi(h) = \frac{1}{27^{d(h)+1}}\frac{1}{2^{27^d}}$, where $\frac{1}{27^{d(h)+1}}$ distributes the confidence budget δ to H_d, and $\frac{1}{2^{27^d}}$ distributes the confidence budget uniformly within a given H_d.

Thus, replacing $M = 2^{27^d}$ in **Theorem 3.2** we have the following for all $h \in H_d$:

$$P(L(h) \leq \hat{L}(h,S) + \sqrt{\frac{ln(2^{27^d})/\delta}{2n}}) \geq 1 - \delta \tag{5}$$

3.2

For all $h \in H$ we take **Theorem 3.5**. However, our $\pi(h)$ is distributed as mentioned above:

$$P(L(h) \leq \hat{L}(h,S) + \sqrt{\frac{ln(2^{27^{d}(h)}27^{d(h)+1})/\delta}{2n}}) \geq 1 - \delta \tag{6}$$

3.3

Examining the probability bounds eq. 5 and eq. 6 we can see that the value of d will affect the number of hypothesis $M = 2^{27^d}$ in each H_d, the probability confidence interval $\sqrt{\frac{ln(2^{27^d})/\delta}{2n}}$ for each H_d directly proportionately. On the other hand, it is inversely proportional to the confidence budget each H_d ($\frac{1}{2^{d(h)+1}}$) is given.

Consequently, a small value of d will have fewer hypotheses, tighter bound, and higher confidence for each H_d.

Contrarily, a large value of d will have more hypotheses, looser bound, and less confidence for each H_d.

We can intuitively understand the decision between a small and large value of d as being a question of under- vs overfitting. The model will be unable to capture enough information with a small d, thus underfitting. With a large value of d, it will capture meaningless noise and instead overfit.

4 Kernels

4.1 Distance in feature space

We know that distance between x and y is defined as $d(x, y) = \|x - y\|$. So, in our case this

$$d(\Phi(x), \Phi(y)) = \|\Phi(x) - \Phi(y)\| \tag{7}$$

We also know that in RKHS the norm $\|f\| = \sqrt{\langle f, f \rangle}$.

We also use the property that

$$\langle \Phi(x), \Phi(x') \rangle = \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$$

Thus eq. 7 can be rewritten as:

$$\|\Phi(x) - \Phi(y)\| = \sqrt{\langle \Phi(x) - \Phi(y), \Phi(x) - \Phi(y) \rangle}$$

$$= \sqrt{\langle k(x, \cdot) - k(z, \cdot), k(x, \cdot) - k(z, \cdot) \rangle} = \sqrt{k(x, x) - k(x, z) - k(x, z) + k(z, z)}$$

$$= \sqrt{k(x, x) - 2k(x, z) + k(z, z)}$$

$$q.e.d$$

4.2 Sum of kernels

We have matrix K_1, where $K_{1i,j} = k_1(x_i, x_j)$

We also have matrix K_2, where $K_{2i,j} = k_2(x_i, x_j)$

We then have K, where $K_{i,j} = k(x_i, x_j) = k_1(x_i, x_j) + k_2(x_i, x_j) = K_{1i,j} + K_{2i,j}$

To prove k is p.d. we have to prove K is p.d. For this we prove the following:

$$\forall v \in \mathbf{R}^m : v^T K v \geq 0 \implies \sum_{i,j} v_i K_{i,j} v_j \geq 0$$

$$\implies \sum_{i,j} v_i \langle x_i, x_j \rangle v_j \geq 0 \implies \sum_j \langle \sum_i v_i x_i, x_j \rangle v_j \geq 0$$

$$\implies \langle \sum_i v_i x_i, \sum_j v_j x_j \rangle \geq 0$$

Using the property:

$$\langle f, f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

the dot product can be rewritten as $\langle y, y \rangle$

$$\implies \langle y, y \rangle \geq 0$$

This is 0 *iff.* v is 0. Else it is ≥ 0. Thus K is p.d. and $k = k_1 + k_2$ is p.d. *q.e.d.*

4.3 Rank of gram matrix

We denote $A =$ input data matrix, $shape(A) = d \times m$, meaning the rows are the features and the columns are the entities. Then $rank(A_{d \times m}) \leq min(m, d)$.

We have K gram matrix where $K_{i,j} = k(x_i, x_j)$. Since k is the linear kernel $k(x, z) = x^T z$ then $K = A^T A$.

We try to prove that $rank(K) = rank(A^T A) = rank(A)$

We use the nullspace of A. We have $v \in N(A)$, where $N(A)$ is nullspace of A.

Then $Av = 0 \implies A^T Av = 0 \implies v \in N(A^T A)$. So

$$N(A^T) \subseteq N(A^T A) \tag{8}$$

We then choose $v \in N(A^T A)$.

Then $A^T Av = 0 \implies v^T A^T Av = 0 \implies (Av)^T (Av) = 0$. We know that for any vector y, $y^T y = \|y^2\|$. Then $\|(Av)\|^2 = 0 \implies Av = 0 \implies v \in N(A)$. So $N(A^T A) \subseteq N(A)$.

But since we already have eq. 8, this must mean that $N(A^T A) = N(A) \implies dim(N(A^T A)) = dim(N(A)) \implies rank(A^T A) = rank(A) \leq min(m, d)$