1: PAC Learning

Rule 1: You are free to combine any of the parts as they are.

Rule 2: You may also cut any of the parts into two distinct pieces before using them.

(1a)

Given N parts, each product that can be made out of these parts is a distinct hypothesis h in the hypothesis space H. From *Rule 1*, a worker can choose to include or not include any of the parts in a product. This can be viewed as a monotone conjunction as a product is defined by choosing to include or not include each of the N parts. There exists 2^N possible products as there are two choices for each of the N parts. We will not consider the product constructed by using none of the parts.

$\boxed{|H| = 2^N - 1}$

(1b)

The experienced worker now creates a product using *Rule 1* and *Rule 2*. There are now four choices that can be made for each of the parts: don't include it, include it, cut the part and use the first half or cut the part and use the second half. A product is now defined as making four choices for each of the N parts. Thus there are 4^N possible products. We will not consider the product constructed by using none of the parts.

$\boxed{|H| = 4^N - 1}$

(1c)

By applying the principles of Occams's Razor we can make a statement about the number of required examples the robot will have to see to have an error of 0.01 with probability 99% on products with 6 available parts.

Given a hypothesis space H, we can say with probability $1 - \delta$, a hypothesis $h \in H$, that is consistent with a training set of size m, will have an error $< \epsilon$ on future examples if

$$m > \frac{1}{\epsilon}(ln(|H|) + ln\frac{1}{\delta})$$

We want an error rate of $\epsilon = 0.01$ with probability $1 - \delta = 0.99$ with a $|H| = 4^6 = 4,096$.

$$m > \frac{1}{0.01}(ln(4,095) + ln\frac{1}{0.01})$$
$$m > 1,292.27$$

The robot will have to see at least $1,293$ examples to guarantee a 0.01 error with probability 99% if there are 6 available parts. We round up as the number of required examples must be an integer value and rounding down would not satisfy the equality.

$\boxed{\text{at least 1,293 examples}}$

(2)

2: VC Dimensions

(1) We want to prove that a finite hypothesis space $\mathcal{C}$ has a VC dimension at most $log_2|\mathcal{C}|$. That is, $VC(\mathcal{C}) \leq log_2|\mathcal{C}|$.

Proof by contradiction: Assume the opposite that $VC(\mathcal{C}) > log_2|C|$ is true.

Shatter: A set of examples S is *shattered* by a set of functions H if for every partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples.

VC Dimension: The *VC Dimension* of hypothesis space H over instance space X is the size of the largest finite subset of X that is shattered by H.

Take a set of examples S that is the largest finite subset of X that is shattered by a hypothesis space $\mathcal{C}$ and is of size d. There exists 2^d (the number of possible binary vectors of length d) ways to partition S. Thus $|\mathcal{C}| = 2^d$ as we have 2^d hypothesis functions.

$$VC(\mathcal{C}) > log_2|C|$$

The VC Dimension of $\mathcal{C} = |S| = d$ as S is the largest finite subset of X that is shattered by H. The size of the hypothesis class is 2^d.

$$d > log_2(2^d)$$

$$d > d$$

Arriving at the contradiction $d > d$. Thus proving $VC(\mathcal{C}) \leq log_2|\mathcal{C}|$.

(2a)

(2b)

(3)

(4)

(5)

3: AdaBoost