Question 1.

Consider the Winnow algorithm.

(a) What concept class was Winnow designed for? What is Winnow's mistake bound for that class?

(b) Adapt the algorithm to learn general conjunctions. How will the mistake bound change?

Answer:

(a) Winnow was designed for monotone k-disjunctions, for which it *dramatically* improves the mistake bound.

$$\text{MB} = 2 + 2k \lg n$$

(b) Winnow only works for linearly separable concepts. Conjunctions are not linearly separable, however, monotone conjunctions are.

We can make conjunctions monotone by the following *basis expansion*: For each $x_i \in \{0,1\}^n$, create $x_i' = [x, \neg x] \in \{0,1\}^{2n}$. Run the algorithm on examples x_i' while again negating the labels as in question (c) about generalization algorithm to turn the conjunction into a disjunction that Winnow is designed for.

The mistake bound changes as follows:

$$2 + 2k \lg 2n = 2 + 2k(1 + \lg n) = 2 + 2k \lg n + 2k$$

i.e., only by an additive constant $2k$.

Question 2.

Consider the halving algorithm with hypothesis class (initial hypothesis) $\mathcal{H}_1$ of all non-contradictory conjunctions on 3 propositional variables.

(a) Determine $|\mathcal{H}_1|$.

(b) Give an upper bound on $|\mathcal{H}_2|$ given that first prediction was incorrect.

Answer:

(a) $3^3 = 27$ (Each of the 3 variables may be absent, positive, or negative in the conjunction.)

(b) The halving algorithm decides by a majority vote so at least $\lceil \frac{27}{2} \rceil = 14$ hypotheses in $\mathcal{H}$ were inconsistent with the observation; those get deleted and at most $\lfloor \frac{27}{2} \rfloor = 13$ remain.

Question 3.

Consider halving algorithm with the initial version space $\mathcal{H}$ consisting

(a) of all conjunctions of exactly 3 different non-negative literals, i.e.,

$$\mathcal{H} = \{\, p_i \wedge p_j \wedge p_k \mid 1 \leq i < j < k \leq n \,\}$$

(b) of all conjunctions that use some of the given variables (and the empty conjunction).

(c) of all n-CNFs.

1. For each scenario, determine if the learner learns $\mathcal{H}$ online (in the mistake-bound model) and justify your answer.

   Answer:

   The halving algorithm makes at most $\lg|\mathcal{H}|$ mistakes when learning a hypothesis from $\mathcal{H}$.

   (a) $|\mathcal{H}| = \binom{n}{3} \leq n^3$, so $\lg|\mathcal{H}| \leq 3\lg n \leq \text{poly}(n)$. Hence, the learner learns $\mathcal{H}$ online.

   (b) $|\mathcal{H}| = 2^{2n}$, so $\lg|\mathcal{H}| = 2n$ is polynomial in n and the algorithm learns $\mathcal{H}$ online.

   (c) $|\mathcal{H}| = 2^{\sum_{i=1}^{n}\binom{n}{i}2^i} = 2^{3^n-1}$, so $\lg|\mathcal{H}| = 3^n - 1$. Hence, the algorithm does not learn $\mathcal{H}$ online.

2. For each scenario where the learner learns in the mistake bound model, decide if they learn efficiently as well. Assume that checking the consistency of a single hypothesis with an observation takes a unit of time.

   Answer:

   (a) $|\mathcal{H}| \leq \text{poly}(n)$, so yes.

   (b) $|\mathcal{H}|$ is super-polynomial in n, hence no.

3. For the **first** case, assume the first example is $(0, 1, 1, 1, \ldots 1)$ and it is a negative instance. What will be the learner's prediction for the second example, which is $(0, 1, 0, 1, \ldots)$? Justify your answer.

   Answer:

   On the first observation, all conjunctions not containing p_1 vote for a positive label. There are $\binom{n-1}{3}$ of those and they all will be deleted on the first hypothesis update. We are left with $\binom{n-1}{2}$ hypotheses that contain p_1.

   Since all hypotheses contain p_1, they will all vote for a negative label on the second observation. Hence, the prediction will be 0 (a negative label).

Question 4.

Consider the following hypothesis classes. For each hypothesis class, determine its VC dimension and provide a brief proof.

(a) $\mathcal{H} = \{h : \mathbb{R} \mapsto \{0; 1\}, h(x) = [\![x > t]\!], t \in \mathbb{R}\}$

(b) $\mathcal{H} = \{h : \mathbb{R} \mapsto \{0; 1\}, h(x) = [\![t_1 \leq x < t_2]\!], t_1 < t_2 \in \mathbb{R}\}$

(c) $\mathcal{H}$ is the set of all monotone conjunctions on n variables.

Answer:

(a) Consider a point x. For a positive label, we can set t such that $t < x$. For a negative label, we can set t such that $t > x$. Hence, $\text{VC}(\mathcal{H}) \geq 1$.

   Consider two points x_1 and x_2 (without loss of generality $x_1 < x_2$). For labelling $y_1 = 1$ and $y_2 = 0$, we can't shatter the set. Hence, $\text{VC}(\mathcal{H}) < 2$.

   Overall $\text{VC}(\mathcal{H}) = 1$.

(b) Consider two points $x_1 < x_2$. Any possible labelling can be realized by setting $x_1 < t_1 < x_2 < t_2$, $t_1 < x_1 < x_2 < t_2$, $x_1 < t_1 < t_2 < x_2$ or $t_1 < x_1 < t_2 < x_2$. Hence, $\text{VC}(\mathcal{H}) \geq 2$.

   Consider three points $x_1 < x_2 < x_3$. We cannot label $y_1 = y_3 = 1$ while having $y_2 = 0$. Hence, $\text{VC}(\mathcal{H}) < 3$.

   Overall, $\text{VC}(\mathcal{H}) = 2$.

(c) Let us start with the upper bound. There are 2^n monotone conjunctions on n variables. Hence, we have 2^n hypotheses and we cannot shatter more than n elements. Therefore, $\text{VC}(\mathcal{H}) \leq n$.

   As for the lower bound, we need to construct a set of samples shattered by monotone conjunctions. Consider

$$\begin{array}{rcccccc} x_1 = & 0 & 1 & 1 & \ldots & 1 & 1 \\ x_2 = & 1 & 0 & 1 & \ldots & 1 & 1 \\ x_3 = & 1 & 1 & 0 & \ldots & 1 & 1 \\ & & & \vdots & & & \\ x_n = & 1 & 1 & 1 & \ldots & 1 & 0 \end{array} \tag{1}$$

With propositional variables $h_1, h_2, \ldots h_n$, any subset $\{ x_i : i \in I \subseteq \{ 1, 2, \ldots n \} \}$ of the above sample set is isolated with the monotone conjunction $\bigwedge_{i \in \{ 1,2,\ldots n \} \setminus I} h_i$.

For example, the monotone conjunction $h_2 \wedge h_3$ picks the elements $\{ x_1, x_4, \ldots, x_n \}$. The empty conjunction (when $I = \{ 1, 2, \ldots n \}$) is a tautology and thus picks all the samples. And when $I = \emptyset$, then we have the conjunction $\bigwedge_{i=1}^{n} h_i$, which does not select any samples.

Hence, the sample set is indeed shattered and we have $\mathrm{VC}(\mathcal{H}) \geq n$.

Overall, it holds that $\mathrm{VC}(\mathcal{H}) = n$.