Question 1. (5 points)

Consider classification with $Y = \{ 0, 1 \}$, where $P_c(y^*|x)$ is the probability that y^* is the true class of x, and rewards are given as

$$r_k = \begin{cases} 0 \text{ if } y = y^* \\ -1 \text{ if } y = 0 \text{ and } y^* = 1 \\ -3 \text{ if } y = 1 \text{ and } y^* = 0 \end{cases}$$

Consider the policy

$$y(x) = \arg\max_y P_c(y|x)$$

is this policy necessarily optimal, i.e. does it always coincide with the policy

$$\tilde{y}(x) = \arg\max_y \mathbb{E}\left(r|x, y\right)$$

? Justify your answer mathematically.

Answer:

No. Consider an observation x such that $P_c(0|x) = 1/3$ and $P_c(1|x) = 2/3$

$$\mathbb{E}\left(r|x, 0\right) = -1 \cdot P(-1|x, 0) - 3 \cdot P(-3|x, 0) = -1 \cdot P_c(1|x) - 3 \cdot 0 = -2/3$$
$$\mathbb{E}\left(r|x, 1\right) = -1 \cdot P(-1|x, 1) - 3 \cdot P(-3|x, 1) = -1 \cdot 0 - 3 \cdot P_c(0|x)) = -1$$

So

$$0 = \arg\max_y \mathbb{E}\left(r|x, y\right) \neq \arg\max_y P_c(y|x) = 1$$

Question 2. (2 points)

Discuss how the *exploration-exploitation* dilemma manifests itself in the *concept learning* scenario. Specify the conditions on which the execution of random actions would (would not) be useful for a concept-learning agent.

Answer:

In concept learning (as defined in the lectures), the reward r for x, y is known with certainty on the first execution of y one step after receiving x so getting more reward samples for that pair is useless for estimating $P(r|x, y)$. However, without further assumptions, future observation may depend on the current reward so random (non-optimal) actions could be used to explore such a dependence.

Consider e.g. a "husband agent" told by "wife environment": $x =$"wash the dishes". The agent does not like to receive such x because the rewards for acting on such x (which include time spent, missing other joys, ..) are usually small. A possible strategy is to break all the dishes during wash-up, receiving a very low r this time but with the chance of never receiving x again.

On the other hand, when x are sampled i.i.d., so they do not depend on the history, exploration through randomized actions is pointless.

Question 3. (2 points)

Consider an algorithm that learns *monotone* disjunctions (or monotone conjunctions) from n-tuples of Boolean attribute values corresponding to n propositional variables. How can you use that algorithm to learn *general* disjunctions (or conjunctions) without changing it? You may change the number of inputs. How will your solution change the mistake bound in the case of the Winnow algorithm? Consider the number s of literals in the target disjunction constant.

Answer:

By basis expansion: introduce additional n attributes holding the inverted values of the original n attributes, thus converting the task to learning a monotone disjunction on $2n$ variables from $2n$-tuples.

For Winnow, the bound

$$2 + 2s \log n$$

changes to

$$2 + 2s \log 2n = 2 + 2s \log 2n = 2 + 2s(1 + \log n) = 2 + 2s \log n + 2s$$

i.e., only by the additive constant $2s$.

Question 4. (1 points)

Let h, h' be propositional conjunctions. Is $h' \models h$ equivalent to $h \subseteq h'$? Justify your answer.

Answer:

The two relations are not equivalent. For example $\mathrm{p} \wedge \neg \mathrm{p} \models \mathrm{q}$ but $\mathrm{q} \not\subseteq \mathrm{p} \wedge \neg \mathrm{p}$. The equivalence would hold if h' was not tautologically false.

Question 5. (4 points)

Let h, h' be contingent propositional conjunctions that prescribe policies by

$$y = h(x) = 1 \text{ iff } x \models h$$

We say that h at least as general as h' if $h(x) = 1$ for any $x \in X$ such that $h'(x) = 1$. Is it true that $h' \models h$ if and only if h is at least as general as h'? Justify your answer.

Answer:

No. Consider

$$h = \mathrm{p}$$
$$h' = \mathrm{q}$$
$$X = \{ \mathrm{p} \wedge \mathrm{q} \}$$
$$h(\mathrm{p} \wedge \mathrm{q}) = h'(\mathrm{p} \wedge \mathrm{q}) = 1$$

h is as general as h' but $h' \not\models h$.

Tautological consequence does not depend on an interpretation domain, while the generality relation depends on the observation set X.