Q1-1: Consider Instance space *X*, Hypothesis space *H*, training examples *D*. Suppose that every *h* in our version space $VS_{H, D}$ is consistent with *m* training examples. Which of the following statement(s) is/are TRUE?

A. *The version space $VS_{H, D}$ is ε-exhausted with respect to c and D if every hypothesis h in $VS_{H, D}$ has training error < ε.*

B. *"The probability that $VS_{H, D}$ is not ε-exhausted ≤ $|H|e^{-\varepsilon m}$" provides a bound for the probability that ANY learner will output a hypothesis h with error > ε.*

1. Both the statements are TRUE.
2. Statement A is TRUE, but statement B is FALSE.
3. Statement A is FALSE, but statement B is TRUE.
4. Both the statements are FALSE.

Q1-1: Consider Instance space *X*, Hypothesis space *H*, training examples *D*. Suppose that every *h* in our version space $VS_{H, D}$ is consistent with *m* training examples. Which of the following statement(s) is/are TRUE?

A. *The version space $VS_{H, D}$ is ε-exhausted with respect to c and D if every hypothesis h in $VS_{H, D}$ has training error < ε.*

B. *"The probability that $VS_{H, D}$ is not ε-exhausted ≤ $|H|e^{-\varepsilon m}$" provides a bound for the probability that ANY learner will output a hypothesis h with error > ε.*

1. Both the statements are TRUE.
2. Statement A is TRUE, but statement B is FALSE.
3. Statement A is FALSE, but statement B is TRUE.
4. Both the statements are FALSE. ⬅

The version space $VS_{H, D}$ is ε-exhausted with respect to c and D if every hypothesis h is $VS_{H, D}$ has **true** error < ε

"The probability that $VS_{H, D}$ is not ε-exhausted ≤ $|H|e^{-\varepsilon m}$" provides a bound for the probability that ANY **consistent** learner will output a hypothesis h with error > ϵ.

Q1-2: We know that Probability that the version space is not ε-exhausted after *m* training examples is at most $|H|e^{-\varepsilon m}$ where symbols have their usual meanings. From this we can derive that if $error_{train}(h) = 0$, *then with probability at least* $(1 - \delta)$, $error_{true}(h) \leq 1/m * A$ *where A is*

1. ln (|H|) + ln (δ)
2. ln (|H|) + ln (1/δ)
3. ln (1/|H|) + ln (δ)
4. ln (1/|H|) + ln (1/δ)

Q1-2: We know that Probability that the version space is not ε-exhausted after *m* training examples is at most $|H|e^{-\varepsilon m}$ where symbols have their usual meanings. From this we can derive that if *$error_{train}(h) = 0$, then with probability at least $(1 - \delta)$, $error_{true}(h) \leq 1/m * A$ where A is*

1. $\ln(|H|) + \ln(\delta)$
2. $\ln(|H|) + \ln(1/\delta)$ ⟵
3. $\ln(1/|H|) + \ln(\delta)$
4. $\ln(1/|H|) + \ln(1/\delta)$

We can write the bound as follows:

$Pr[(\exists h \text{ in } H) \text{ s.t. } (error_{train}(h) = 0) \text{ AND } (error_{true}(h) > \varepsilon)] \leq |H|e^{-\varepsilon m} = \delta$

Hence, with at least probability $(1 - \delta)$, $error_{true}(h) \leq \varepsilon$, where ε can be obtained by solving $|H|e^{-\varepsilon m} = \delta$.

Q2-1: Which of the following statement(s) is/are TRUE?

A. *PAC analysis formalizes the learning task and allows for non-perfect learning.*

B. *Finding a consistent hypothesis is easier for smaller concept classes.*

C. *In PAC analysis, we are trying to bound training error of a hypothesis.*

1. True, True, True
2. True, False, False
3. True, False, True
4. False, False, False

Q2-1: Which of the following statement(s) is/are TRUE?

A. *PAC analysis formalizes the learning task and allows for non-perfect learning.*

B. *Finding a consistent hypothesis is easier for smaller concept classes.*

C. *In PAC analysis, we are trying to bound training error of a hypothesis.*

1. True, True, True
2. True, False, False
3. True, False, True
4. False, False, False

Q2-2: Similar to “PAC analysis example: learning decision trees”, consider the case when each instance has *n* features and each feature has 3 possible values. Let learned hypotheses are DTs of depth 2 using only 2 variables. What is |H|? Note: nC2 = $\binom{n}{2}$

1. nC3 x 2^6
2. nC3 x 2^9
3. nC2 x 2^9
4. nC2 x 2^6

Q2-2: Similar to “PAC analysis example: learning decision trees”, consider the case when each instance has *n* features and each feature has 3 possible values. Let learned hypotheses are DTs of depth 2 using only 2 variables. What is |H|? Note: nC2 = $\binom{n}{2}$

1. nC3 x 2^6
2. nC3 x 2^9
3. nC2 x 2^9 ⟵
4. nC2 x 2^6

#possible split choices (choose 2 features out of n) = nC2
#leaves = 9. Each leaf has 2 possible labellings. So, #possible leaf labellings = 2^9

Q3-1: Which of the following statement(s) is/are TRUE?

A. *Agnostic PAC learnability requires learning a hypothesis with true error at most ε*
B. *In the case of agnostic PAC learning, if $error_{train}(h_{best}) = 0$, then with probability at least $(1 - \delta)$, $error_{true}(h_{best}) \leq (1/2m)\ sqrt[\ln |H| + \ln(1/\delta)]$*

1. True, True
2. True, False
3. False, True
4. False, False

Q3-1: Which of the following statement(s) is/are TRUE?

A. *Agnostic PAC learnability requires learning a hypothesis with true error at most ε*

B. *In the case of agnostic PAC learning, if $error_{train}(h_{best}) = 0$, then with probability at least $(1 - \delta)$, $error_{true}(h_{best}) \leq (1/2m)\ sqrt[\ln |H| + \ln(1/\delta)]$*

1. True, True
2. True, False
3. False, True
4. False, False ⬅

Being agnostic PAC learnable is a stronger condition, since for all distributions, we can get close to the optimal error.

We want to get the expression for ε in agnostic PAC learning case. Solve for ε using:

$$m \geq \frac{1}{2\varepsilon^2}\left(ln|H| + ln\left(\frac{1}{\delta}\right)\right)$$

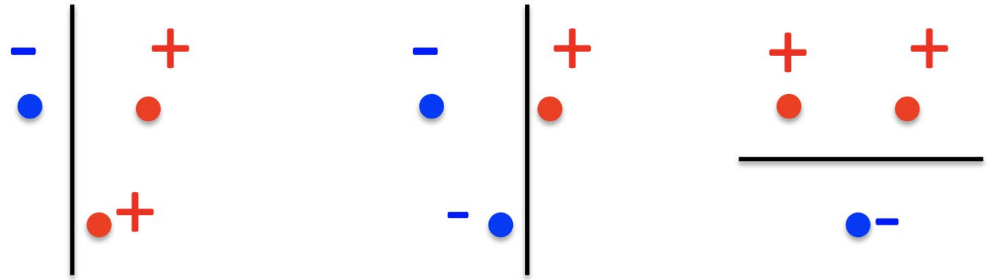
Q3-2: What’s the VC dimension of a decision stump in **R^2**? A decision stump is of the form f(x) = sign(x_i > b) or f(x) = sign(x_i < b) for one feature x_i and some bias parameter b.

1. VC = 1
2. VC = 2
3. VC = 3
4. VC = 4

Q3-2: What's the VC dimension of a decision stump in $\mathbf{R}^2$? A decision stump is of the form f(x) = sign(x_i > b) or f(x) = sign(x_i < b) for one feature x_i and some bias parameter b.

1. VC = 1
2. VC = 2
3. VC = 3
4. VC = 4

VC ≥ 3: All possible labelings of 3 points can be shattered.



VC = 3: For all placements of 4 points, there exist a labeling that can't be shattered.