Probably Approximately Correct (PAC) learning framework

PAC is a framework for mathematical analysis of machine learning [wiki]. It helps define the class of learnable concepts in terms of the number of sample points needed to achieve an approximate solution, sample complexity and the time and space complexity of the learning algorithm. These depends on the cost of the computational representation of the concepts.

Basic definitions and notation:

- X ... set of all possible examples or instances → "input space"
- Y ... set of all possible labels or target values → "output space": for now $Y = \{0, 1\}$.
- A concept $c: X \to Y$. Since $Y = \{0, 1\}$ we can identify c with subset of X that maps to 1 with the indicator function of that subset. For this reason, we sometimes refer to concept directly as that subspace of X.
- A concept class is a set of concepts we may wish to learn, denoted by C.
- We assume that examples are independantly and identically distributed (i.i.d) according to some fixed but unknown distribution D.

Learning problem:

Learner considers a fixed set of possible concepts H, called a hypothesis set, which may not comcide with C. He receives a sample $S = (x_1, \dots, x_m)$ drawn i.i.d. according to D as well as the labels $(c(x_1), \dots, c(x_m))$, based on specific target concept $c \in C$ to learn. His task is to use labeled sample S to select a hypothesis $h_S \in H$ that has small generalization error with respect to concept c.

Definition 1: Generalization error.

Given a hypothesis $h \in H$, a target concept $c \in C$ and an underlying distribution D, the generalization error or risk of h is defined as

$$R(h) = \Pr_{x \sim D}[h(x) \neq c(x)] = \mathop{E}_{x \sim D}[1_{h(x) \neq c(x)}],$$

where 1_ω is the indicator function of event ω.

Since learner does not have access to D and c, he can not measure true error. That is why we introduce emperical error of h on the labeled sample S.

Remark:

Probability of event $h(x) \neq c(x)$ depends on distribution D of our examples. Our notation makes this explicit with writing $\Pr_{x \sim D}[\cdot]$ and $\mathop{E}_{x \sim D}[\cdot]$ → mean value of $f(x)$, given x is i.i.d. with D.

Definition 2: Emperical error

Given a hypothesis $h \in H$, a target concept $c \in C$ and a sample $S = (x_1, \dots, x_m)$, the emperical error or emperical risk of h is defined by

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}$$

Remarks:

1.) Emperical error of $h \in H$ is its average error over the sample S, while generalization error is its expected error based on distribution D.

2.) With 1.) in mind it is obvious that, for a fixed $h \in H$, the expectation of the emperical error based on i.i.d. sample is equal to the generalization error

$$\underset{S \sim D^m}{E}\left[\hat{R}(h)\right] = R(h)$$

, to see this on technical level

$$\underset{S \sim D^m}{E}\left[\hat{R}(h)\right] \overset{\text{linearity of E[·]}}{=} \frac{1}{m} \sum_{i=1}^{m} \underset{S \sim D^m}{E}\left[1_{h(x_i) \neq c(x_i)}\right] = \frac{1}{m} \sum_{i=1}^{m} \underset{x_i \sim D}{E}\left[1_{h(x_i) \neq c(x_i)}\right] \quad (*)$$

equal to $x_1 \sim D$, $x_2 \sim D$, $\dots$, $x_m \sim D$

since examples are i.i.d.

we get the same result for each x_i

$$\Rightarrow (*) = \underset{x \sim D}{E}\left[1_{h(x) \neq c(x)}\right] = R(h)$$

(PAC prediction) by hypothesis class $H \rightarrow$ in proper PAC framework $C = H$

Definition 3: PAC-learning

A concept class C is said to be PAC-learnable if there exists an algorithm A and a polynomial function $\text{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon, \delta \in (0,1)$, for all distributions D on X and for any target concept $c \in C$, the following holds for any sample size $m \geq \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c)\right)$: ($h_s \in H$ is output of A based on sample S)

$$\Pr_{S \sim D^m}\left[R(h_s) \leq \varepsilon\right] \geq 1 - \delta$$

, where with $O(n)$, we denote an upper bound on the cost of the computational representation of any $x \in X$ and by $\text{size}(c)$ the maximal cost of the computational representation of $c \in C$.

approximately correct (small error) — with high probability — PAC.

If A further runs in $\text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c)\right)$, then C is said to be efficiently PAC-learnable. When such A exists, it is called a PAC-learning algorithm for C.

Remarks:

- other possible definitions in the literature ⚠ ($h_s \in H$, not C for example, ...)
- if running time of the algorithm is polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta} \Rightarrow m$ must be polynomial if the full sample is received by the algorithm.
- distribution free model: no assumption about D
- training sample and the test examples used for defining error are drawn with same
- PAC is dealing with learnability for a concept class C and not some $c \in C$.
- algorithm knows C (or H in PAC prediction framework).
- polynomial dependancy on n and $\text{size}(c)$ is often omitted.
- $1-\delta$ is usually called confidence and $1-\varepsilon$ is called accuracy.
- when X is finite, $C \subseteq H$

Guarantees for finite hypothesis sets – consistent case

Definition 4: Consistent hypothesis

The hypothesis h is said to be consistent if its error on the training sample is 0.

Theorem 1: Learning bounds – finite H, consistent case

Let H be a finite set of functions mapping from X to Y. Let $\mathcal{A}$ be an algorithm that for any target concept $c \in H$ and i.i.d. sample S returns a consistent hypothesis h_S: $\hat{R}(h_S) = 0$. Then, for any $\varepsilon, \delta \in (0,1)$, the inequality $\Pr_{S \sim D^m}[R(h_S) \leq \varepsilon] \geq 1 - \delta$ holds if

$$\boxed{m \geq \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right)}$$

Equivalent: for any $\varepsilon, \delta \in (0,1)$, we can expect $R(h_S) \leq \frac{1}{m}(\log|H| + \log\frac{1}{\delta})$ with probability at least $1-\delta$.

Proof: Theorem 1

Fix $\varepsilon \in (0,1)$. We do not know which consistent hypothesis $h_S \in H$ is selected by $\mathcal{A}$. Therefore we need to give a bound that holds for the set of all consistent hypothesis. Thus, we will bound the probability that some $h \in H$ would be consistent and have error more then ε.

$$\Pr[\exists h \in H : \hat{R}(h) = 0 \wedge R(h) > \varepsilon] =$$

$$= \Pr\left[\bigcup_{h \in H} (\hat{R}(h) = 0 \wedge R(h) > \varepsilon)\right] \leq$$

$$\leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \wedge R(h) > \varepsilon] \leq \quad \text{(union bound, easily proven by induction)}$$

$$\leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \mid R(h) > \varepsilon] \quad (1) \quad \text{(definition of conditional probability } P(A|B) = \frac{P(A \cap B)}{P(B)}\text{)}$$

Now, consider any $h \in H$ with $R(h) > \varepsilon$. Then, the probability that h would be consistent on a training sample S drawn i.i.d. can be bounded as:

$$\Pr_{S \sim D^m}[\hat{R}(h) = 0 \mid R(h) > \varepsilon] \leq (1-\varepsilon)^m \quad (2) \quad \text{, since}$$

$$R(h) = \Pr_{x \sim D}[h(x) \neq c(x)] > \varepsilon \Rightarrow \Pr_{x \sim D}[h(x) = c(x)] = 1 - \Pr_{x \sim D}[h(x) \neq c(x)] < 1 - \varepsilon$$

This has to hold for all m points of the S at the same time, giving us (2).

Using (2) in (1) gives us:

$$\Pr[\exists h \in H \cdot \hat{R}(h) = 0 \wedge R(h) > \varepsilon] \leq |H|(1-\varepsilon)^m \leq |H| e^{-\varepsilon m}$$

Since target concept $c \in H$, we always have at least one consistent hypothesis:

$$\Pr[\exists h \in H \cdot \underbrace{\hat{R}(h) = 0}_{(*)} \wedge R(h) > \varepsilon] + \Pr[\underbrace{\exists h \in H : \hat{R}(h) = 0 \wedge R(h) \leq \varepsilon}_{\Pr[R(h_S) \leq \varepsilon] \text{ from theorem}}] = 1$$

$\Rightarrow \Pr[R(h_S) \leq \varepsilon] \geq 1 - \delta$ if $\delta \geq (*)$, we take our upper bound for $(*)$ from before

$$\delta \geq |H| e^{-\varepsilon m} \Rightarrow \boxed{m \geq \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right)}$$

■

Remarks:

- when hypothesis set H is finite, and not superexponentials dependent on n or ϵ, a consistent algorithm A is a PAC-learning algorithm (sample complexity (m) is dominated by polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$)
- generalization error of consistent hypotheses is upper bounded by a term $\frac{1}{m}$.
- price to pay to come up with consistent algorithm is the use of larger H, containing target concepts
- upper bound increases as $\log|H|$, which can be interpreted as the number of bits needed to represent H $(\log_2|H|)$.

Guarantees for finite hypothesis sets - inconsistent case

In practice, it is typical that learning problems are too difficult or the concept classes more complex than the hypothesis set used by the learning algorithm. For that reason, the hypothesis set often does not include hypothesis, consistent with labeled sample.

Theorem 2: Markov inequality

If X is a non-negative random variable and $a > 0$, then:

$$\Pr[X \geq a] \leq \frac{E[X]}{a}$$

Proof: Theorem 2

$$E[X] = \int_{-\infty}^{\infty} x f(x)\,dx \text{, } X \text{ non-negative} \Rightarrow E[X] = \int_{0}^{\infty} x f(x)\,dx$$

$$E[X] = \int_{0}^{a} x f(x)\,dx + \int_{a}^{\infty} x f(x)\,dx \geq \int_{a}^{\infty} x f(x)\,dx \geq a \int_{a}^{\infty} f(x)\,dx = a \Pr[X \geq a]$$

$$\Rightarrow \Pr[X \geq a] \leq \frac{E[X]}{a} \quad \blacksquare$$

Lemma 1: Hoeffding's lemma

Let X be a random variable with $E[X] = 0$ and $a \leq X \leq b$ with $b > a$. Then, for any $t > 0$, the following inequality holds:

$$E[e^{tX}] \leq e^{\frac{1}{8}t^2(b-a)^2}$$

Proof: Lemma 1

By definition of convexity of map $x \longmapsto e^x$, for all $x \in [a, b]$:

$$e^{tx} \leq \frac{b-x}{b-a} e^{ta} + \frac{x-a}{b-a} e^{tb}$$

By using $E[X] = 0$:

$$E[e^{tX}] \leq E\left[\frac{b-X}{b-a} e^{ta} + \frac{X-a}{b-a} e^{tb}\right] = \frac{b}{b-a} e^{ta} + \frac{-a}{b-a} e^{tb} = e^{\phi(t)}, \text{ where}$$

$\phi(t) = ta + \log\left(\frac{b}{b-a} + \frac{-a}{b-a} e^{t(b-a)}\right)$. For any $t > 0$, $\alpha := \frac{-a}{b-a}$:

$$\phi'(t) = a - \frac{a}{\frac{b}{b-a} e^{-t(b-a)} - \frac{a}{b-a}} \quad \text{and} \quad \phi''(t) = \frac{\alpha}{[(1-\alpha)e^{-t(b-a)} + \alpha]} \frac{(1-\alpha)e^{-t(b-a)}}{[(1-\alpha)e^{-t(b-a)} + \alpha]} (b-a)^2$$

Note: $\phi(0) = \phi'(0) = 0$ and $\phi''(t) = v(1-v)(b-a)^2$, where $v = \frac{\alpha}{[(1-\alpha)e^{-t(b-a)} + \alpha]}$.

$v(1-v) \leq \frac{1}{4} \Rightarrow \phi''(t) \leq \frac{1}{4}(b-a)^2$. Then by Taylor's series with explicit form of reminder $\left(R_k(x) = \frac{f^{(k+1)}(\xi_L)}{(k+1)!}(x-a)^{k+1},\ \xi_L \in [a, x]\right)$ we have $\phi(t) = \phi(0) + t\phi'(0) + \frac{t^2}{2}\phi''(\xi_L) \leq \frac{t^2(b-a)^2}{8}$ for some $\xi_L \in [0, t]$. $\blacksquare$

Theorem 3: Hoeffding's inequality

Let $X_1, \dots, X_m$ be independant real variables with X_i taking values in $[a_i, b_i]$ for all $i \in \{1, \dots, m\}$. Then, for any $\varepsilon > 0$, the following inequalities hold for $S_m = \sum_{i=1}^{m} X_i$:

$$\Pr[S_m - E[S_m] \geq \varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{m}(b_i - a_i)^2}\right)$$
$$\Pr[S_m - E[S_m] \leq -\varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{m}(b_i - a_i)^2}\right)$$

Proof: Theorem 3

We will use the general Chernoff bounding technique for bounding $\Pr[X \geq \varepsilon]$. It consists of following steps:

1.) For any $t > 0$, we use Markov inequality: $\Pr[X \geq \varepsilon] = \Pr[e^{tX} \geq e^{t\varepsilon}] \leq e^{-t\varepsilon} E[e^{tX}]$

2.) Upper bound $g(t)$ is found for $E[e^{tX}]$ (in this case with Lemma 1).

3.) Select t that minimizes $e^{-t\varepsilon} g(t)$.

In our case:

$$\begin{aligned}
\Pr[S_m - E[S_m] \geq \varepsilon] &= \Pr[e^{t(S_m - E[S_m])} \geq e^{t\varepsilon}] \\
&\leq e^{-t\varepsilon} E[e^{t(S_m - E[S_m])}] && \text{Markov inequality} \\
&= \prod_{i=1}^{m} e^{-t\varepsilon} E[e^{t(X_i - E[X_i])}] && \text{Independance of } X_i\text{s} \\
&\leq e^{-t\varepsilon} e^{\frac{1}{8}t^2 \sum_{i=1}^{m}(b_i - a_i)^2} && \text{Lemma 1} \\
&\leq e^{-2\varepsilon^2 / \sum_{i=1}^{m}(b_i - a_i)^2} && \text{Minimization with respect to } t
\end{aligned}$$

Where in last step, we chose $t = \frac{4\varepsilon}{\sum_{i=1}^{m}(b_i - a_i)^2}$ to minimize the upper bound. The second inequality stated in the theorem is shown in a similar way. ∎

Remark:

When the variance $\sigma_{X_i}^2$ of each random variable X_i is known and $\sigma_{X_i}^2$ are relatively small, better concentration bounds can be derived (Bennett's and Bernstein's ineq.).

Corollary 1:

Fix $\varepsilon > 0$ and let S denote i.i.d. sample of size m. Then, for any hypothesis $h : X \to \{0, 1\}$, the following inequalities hold:

$$\Pr_{S \sim D^m}[\hat{R}(h) - R(h) \geq \varepsilon] \leq \exp(-2m\varepsilon^2)$$
$$\Pr_{S \sim D^m}[\hat{R}(h) - R(h) \leq -\varepsilon] \leq \exp(-2m\varepsilon^2)$$

Setting R.H.S. to δ and solving for ε.

Corollary 2: Generalization bound - single hypothesis

Fix a hypothesis $h : X \to \{0, 1\}$. Then, for any $\delta \in (0, 1)$, the following inequality holds:

$$\Pr\left[R(h) \leq \hat{R}(h) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}\right] \geq 1 - \delta$$

Theorem 4: Learning bound - finite case, inconsistent case

Let H be a finite hypothesis set. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds:

$$\forall h \in H: \quad R(h) \leq \hat{R}(h) + \sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}$$

Proof: Theorem 4

Let $h_1, \dots, h_{|H|}$ be elements of H. Using the union bound and applying corollary 2 to each hypothesis yield:

$$\begin{aligned} \Pr\left[\exists h \in H, |\hat{R}(h) - R(h)| > \varepsilon\right] &= \Pr\left[\bigcup_{h \in H} |\hat{R}(h) - R(h)| > \varepsilon\right] \\ &\leq \sum_{h \in H} \Pr\left[|\hat{R}(h) - R(h)| > \varepsilon\right] \\ &\leq 2|H| \exp(-2m\varepsilon^2) = \delta \Rightarrow \varepsilon = \sqrt{\frac{\ln|H| + \ln\frac{2}{\delta}}{2m}} \end{aligned}$$

■

Remarks:

- $R(h) \leq \hat{R}(h) + O\left(\sqrt{\frac{\log_2|H|}{m}}\right)$, $\log_2|H| \sim$ nb. of bits needed to represent H
- larger sample size m guarantees better generalization
- bound is less favourable function of $\frac{\log|H|}{m} \Rightarrow$ for fixed $|H|$ we need quadratically (up to a constant factor) larger labeled sample to achieve the same guarantee as in consistent case
- bound seggests a trade-off between reducing empirical error versus controlling the size of hypothesis set.

Generalizations

1. Deterministic versus stochastic scenarios

In the most general scenario of supervised learning, the distribution D is defined over $X \times Y$, and the training data is a labeled sample S, drawn i.i.d. according to D:

$$S = ((x_1, y_1), \dots, (x_m, y_m))$$

The learning problem is to find a hypothesis $h \in H$ with a small generalization error

$$R(h) = \Pr_{(x,y)\sim D}[h(x) \neq y] = \mathop{E}_{(x,y)\sim D}[1_{h(x)\neq y}]$$

This scenario is <u>stochastic</u> scenario. It means that output label is a probabilistic function of the input. For ex. if we want to predict gender based on height and weight, then the label will typically not be unique. For each fixed pair, there would be a probability for the label being male or female.
The natural extension of the PAC-learning framework to this setting is known as the agnostic PAC-learning.

<u>Definition 5:</u> Agnostic PAC-learning

Let H be a hypothesis set. A is an agnostic PAC-learning algorithm (returning h_S), if there exists a polynomial function $\text{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon, \delta \in (0,1)$, for all distributions D over $X \times Y$, the following holds for any sample size $m \geq \text{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c))$:

$$\Pr_{S\sim D^m}[R(h_S) - \min_{h\in H} R(h) \leq \varepsilon] \geq 1-\delta$$

If A further runs in $\text{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c))$, then it is said to be an efficient agnostic PAC-learning algorithm.

Remark:

- We get back to deterministic scenario if labels can be uniquely determined by a measurable function $f: X \to Y$. In this case it is enough to consider D over X and obtain training sample by drawing $(x_1, \dots, x_m)$ according to D and label them via f: $y_i = f(x_i)$ for all $i \in \{1, \dots, m\}$

2. Bayes error and noise

In the deterministic case, by definition, there exists a target function f with no generalization error: $R(h) = 0$. In the stochastic case, there is a minimal non-zero error for any hypothesis.

Definition 6: Bayes error

Given a distribution D over $X \times Y$, the Bayes error R^* is defined as the infimum of the errors achieved by measurable functions $h: X \to Y$:

$$R^* = \inf_{h \text{ meas.}} R(h)$$

A hypothesis h with $R(h) = R^*$ is called a Bayes hypothesis or Bayes classifier.

Clearly, the Bayes classifier h_{Bayes} can be defined in terms of conditional probabilities:

$$\forall x \in X, \quad h_{Bayes}(x) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \Pr[y|x]$$

Which are of course determined by D over $X \times Y$.

The average error made by h_{Bayes} on $x \in X$ is thus $\min\{\Pr[0|x], \Pr[1|x]\}$, and this is the minimal possible error. This motivates next definition.

Definition 7: Noise

Given distribution D over $X \times Y$, the noise at point $x \in X$ is defined by

$$\text{noise}(x) = \min\{\Pr[0|x], \Pr[1|x]\}.$$

The average noise or the noise associated to D is $E[\text{noise}(x)]$.

Remarks:

- the average noise in precisely the Bayes error. $E[\text{noise}(x)] = R^*$
- the noise is a characteristic of the learning task indicative of its level of difficulty
- a point with noise close to $\frac{1}{2}$ is sometimes refferd as noisy and is of course a challange for accurate prediction.

3. Estimation and approximation errors

The difference between the error of hypothesis and the Bayes error can be decomposed as:

$$R(h) - R^* = \overbrace{(R(h) - R(h^*))}^{\text{estimation}} + \overbrace{(R(h^*) - R^*)}^{\text{approximation}}$$

where h^* is a hypothesis in H with minimal error, or a best-in-class hypothesis.

→ First term is the estimation error, which is also the basis of agnostic PAC-learning. The estimation error of algorithm $\mathcal{A}$ (est error of h_S returned by $\mathcal{A}$, based on S) can sometimes be in terms of generalization error

→ The second term is approximation error → measures how well can R^* be approximated using H. H is a measure of richness of H. H is never available, since D is unknown. Even estimating the approximation error is difficult.

Examples:

1. PAC learnable concept class of all positive "half-lines"

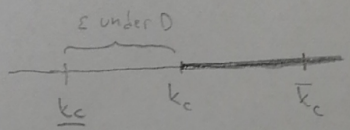
Let our sample space $\mathcal{X}$ be the real line $\mathbb{R}$ and the concept class be the set of all positive "half-lines". This means that every concept c consists of all the points bigger or equal then some real number k_c that characterizes the concept. We label the points $\geq k_c$ with 1 and the ones $< k_c$ with 0. We want to show that this concept class is PAC learnable within proper PAC framework ($H = C$).

Consider an algorithm $\mathcal{A}$ that first, after seeing a training set S which contains m labeled examples $(x_i, c(x_i))$, where $x_i \in \mathbb{R}$ and $c \in C$, selects the biggest number labeled with 0: $\underline{x} \equiv \max_{i : c(x_i)=0} x_i$ and smallest example labeled with 1: $\overline{x} \equiv \min_{i : c(x_i)=1} x_i$.

We know, by construction that $\underline{x} < \overline{x}$. Then $\mathcal{A}$ returns h_S, that is the positive ray corresponding to a point arbitrarily selected from the open interval $(\underline{x}, \overline{x})$.

Let $\varepsilon \in (0,1)$, $k_c \in \mathbb{R}$ lower boundary of true ray c. We define $\overline{k}_c \equiv \max\{k : D([k_c, k)) \leq \varepsilon\}$. So $\overline{k}_c$ is the greatest value of k for which the upper half-open interval $[k_c, k)$ has no more then ε probability weight under the sampling distribution D. Intuitively, $\overline{k}_c$ is exactly (or for discrete probability distributions, as close as we can get to exactly) ε above k_c in "probability distance". We define $R_+ \equiv [k_c, \overline{k}_c]$ and R_- and $\underline{k}_c$ in a symmetric fashion.

Let us now define k_h as the real number marking the lower boundary of the positive ray h. If $k_h \leq \overline{k}_c$, then there will be a probability, under D, of no more then ε that h misclassifies positive example. We want to show that the error of h is less then ε with certain confidence.



We do this by defining the event that $k_h > \overline{k}_c$ as b_+ and the event $k_h < \underline{k}_c$ as b_-. b_+ will only occur if there is no training example $x_i \in R_+$, that is: b_+ will only occur if none of m independent training samples lie inside R_+. Probability for that is at most $(1-\varepsilon)^m \leq e^{-m\varepsilon}$ so $Pr[b^+] \leq e^{-m\varepsilon}$, same for b_-. Note that we have either $k_h < k_c$ or $k_h \geq k_c$ so h may misclassify positive or negative examples, but never both. So if neither b_+ or b_- occur, then probability of h misclassifying an example is $\leq \varepsilon$. Because b_+ or b_- are disjoint, the probability for either to occur $Pr[b_+ \vee b_-] \leq 2e^{-m\varepsilon}$. Putting it all together:

given m independent training examples, we can say, with probability at least $1 - 2e^{-m\varepsilon}$, that error of h is less then ε. So for error to be $\leq \varepsilon$ with confidence at least $1-\delta$, we need $m = \frac{1}{\varepsilon} \ln \frac{2}{\delta}$ training examples, which can be bounded with polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$. $\Rightarrow$ this concept class C is PAC learnable by C.

What is more, algorithm can give result by just going through the training set once, so C is efficiently PAC learnable.

2. Must the target concept class be in the hypothesis set?

The answer to this question is two-fold and can provide some intuition about PAC learnability.

- When X is finite it can be argued that C must be in H. Suppose $|X|=k$ and pick any $c \in C$. PAC learner must work for all distributions D, error ε and confidence $1-\delta$. This gives us an option to choose D as uniform over X, $\varepsilon = \frac{1}{2k}$ and $\delta = \frac{1}{2}$. PAC criterion tells us that it is possible to find a hypothesis h, whose error is less then or equal to $\frac{1}{2k}$. This is less then $\frac{1}{k}$, the probability of a single point under distribution D. So in this case the resulting hypothesis of our algorithm has to agree with c on all points of X, therefore it is identical function to $c \Rightarrow c \in H$ and $C \subseteq H$.
- On the other hand, when X is infinite C does not have to be a subset of H. For instance, consider $X = \mathbb{R}$ and C consists of all positive half-lines. In the hypothesis set, we have only those positive half-lines that begin on rational numbers. Even though, we have only measure-zero fraction of the possible concepts, we can come arbitrarily close to any $c \in C$.

3. Universal concept class: (usage of generalization bound theorem)

Consider $X = \{0,1\}^n$ and let U_n be a concept class formed by all subsets of X. To guarantee a consisten hypothesis the hypothesis class must include the concep class, thus $|H| \geq |U_n| = 2^{2^n}$ (power set of set of all boolean vectors of lenght n). By theorem 1 we know:

$$m \geq \frac{1}{\varepsilon}\left[2^n \log 2 + \log\frac{1}{\delta}\right]$$

The number of training required is exponential in n, which is a cost of the reprcsentati of a point in X. Thus, PAC-learning is not guaranteed by the theorem. In fact, it is not hard to show that this universal concept class is not PAC-learnable.

References:

- original article, introducing PAC learning framework: L.G. Valiant, A Theory of the Learnable, 1984
- chapter 2 of the book: M. Mohri, A. Rostamizadeh and A. Talwalker, Foundations of Machine learning
- princeton lecture notes from course: Computer science 511, Foundations of Machine Learning
  thought by: Rob Schapire, look for scribe notes
- presentation of PAC theory by Chao Zheng from Machine Intelligence Laboratory, Cambridge University Engineering Department