Modeling Tic Tac Toe as a Pac-Learnable problem

→ PAC learnability: A hypothesis class H is PAC learnable if there exists a function $m_H : (0,1) \times (0,1) \longrightarrow \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$, for every distribution D over X, and for every labeling function $g : X \longrightarrow \{0,1\}$, if the realizable assumption holds with respect to H, D, g, then when running the learning algorithm on $m \geq m_H$ independently identically distributed examples generated by D and labeled by g, the algorithm returns a hypothesis h such that with probability of at least $1-\delta$, $L_{D,g}(h) \leq \epsilon$. (43)

For the Tic Tac Toe to work as a PAC learnable problem, we would need some accurate and not that random class of hypothesis classes. Just as a quick experiment, we will use the powerful fact that we know how to forget information without a game state is a win, a loss, a tie and not even a final state to build a learner.

So let the hypothesis class H be the disjunction of some very specific predicates. Let these predicates be:

1. There exists a horizontal line of X's
2. There exists a vertical line of X's
3. There exists a diagonal line of X's
4. There exists two X's together somewhere in the row
5. There exists an X in the center spot of the board

More precisely, assume that H is the class of all possible $h = x_1 \dots x_5$, $x_1, \dots, x_5 = 0$ or 1 such that if $x_i = 1$ then the predicate related to this index will be in the disjunction formula (11001 indicates $P(1) \vee P(2) \vee P(5)$, for example). Furthermore, assume that our goal is only to predict whether a state is a win for the X player or not. We know, by the realizable assumption (and by the knowledge of the simple rules of TicTacToe), that there exists $h_s \in H$ s.t. $L_{D,g}(h_s) = 0$ for any distribution D over the boards of this game and for the labeling function f which take the board x and correctly determines if it's a win or not.

Based on the described problem, assuming that we wanted $P[L_{D,g}(h) \leq \varepsilon = 0,1] \geq 1 - \delta = 0,9$ and respecting the corollary about sample complexity for every finite hypothesis class (37 and 44), we have:

$$m_H(\varepsilon, \delta) = m_H(0,1, 0,1) \leq \left\lceil \frac{\log(|H|/\delta)}{\varepsilon} \right\rceil = \left\lceil \frac{\log(2^5/0,1)}{0,1} \right\rceil = 26$$

Now that we have an upper bound for $m_H(\varepsilon, \delta)$ with $\varepsilon = 0,1$, $\delta = 0,1$, we proceed with our experiment. We will try ERM algorithm (35) on some distributions D and graph the results.

Our first experiment was started with our upper bound $m = 26$ and we ran the program ten thousand times. For this test, we get $P[L_{D,g}(h_S) > 0,1] = 0,035 < 0,1$. Now we try another experiment with $m = 13$ and observe the effects on the error and the probability by passing the $\varepsilon = 0,1$ threshold. With $m = 13$ we get $P[L_{D,g}(h_S) > 0,1] = 0,25 > 0,1$, which shows that we can not obtain the precision and accuracy desired with a number of samples much lower than the upper bound given by the PAC-Learning $m_H(\varepsilon, \delta)$ function. Now with $m = 7$ we get $P[L_{D,g}(h_S) > 0,1] = 0,54 > 0,1$ which is not sufficient for our requirements anymore. (*)

For our next experiment, we are going to try a more general approach: The agnostic PAC-Learnability. We begin with definitions

→ Agnostic PAC-Learnability: A hypothesis class H is said to be Agnostic PAC-Learnable if there exist a function $m_H : (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property: for every $\varepsilon, \delta \in (0,1)$ and every distribution D over $X \times Y$ when running the learning algorithm on $m \geq m_H(\varepsilon, \delta)$ independently identically distributed examples generated by D, the algorithm returns a hypothesis h such that, with probability of at least $1-\delta$ (over the choice of the m training examples), $L_D(h) \leq \min_{h' \in H} L_D(h') + \varepsilon$.

OBS: If we wanted an even more general approach, which is probably the next step of this paper, we would have defined the loss $\mathcal{L}_D(h)$ as $\mathcal{L}_D(h) = E_{Z\sim D}[l(h,z)]$ with some arbitrary loss function $l: H \times Z \to \mathbb{R}_+$ and an arbitrary set Z which we use to take any distribution D.

For us to use the agnostic PAC-learning model though, we will need to guarantee the quality of our samples S. They will need to be ϵ-representative (54). To guarantee these samples on our Tic-Tac-Toe experiment we will define uniform convergence over some hypothesis class H and use two facts:

1. Uniform convergence is sufficient to guarantee ϵ-representative samples by definition. (55)
2. If H is finite, then it enjoys the uniform convergence property and it is agnostic PAC-learnable with sample complexity

$$m_H(\epsilon,\delta) \leq m_H^{UC}(\epsilon/2,\delta) \leq \left\lceil \frac{2\log\left(\frac{2|H|}{\delta}\right)}{\epsilon^2} \right\rceil \quad (57)$$

Now we define ϵ-representative samples as well as uniform convergence:

→ ϵ-representative samples: A training set S is called ϵ-representative with respect to domain Z, hypothesis class H, loss function l and distribution D if $\forall h \in H, |\mathcal{L}_S(h) - \mathcal{L}_D(h)| \leq \epsilon$

→ Uniform convergence: We say that a hypothesis class H has the uniform convergence property with respect to a domain Z and a loss function l if there exists a function $m_H^{UC}: (0,1)\times(0,1) \to \mathbb{N}$ such that for every $\epsilon,\delta \in (0,1)$ and for every probability distribution D over Z, if S is a sample of $m \geq m_H^{UC}(\epsilon,\delta)$ examples drawn identically and independently distributed according to D, then, with probability of at least $1-\delta$, S is ϵ-representative.

We proceed to characterize our experiment:

Hypothesis class H: Disjunction of very simple predicates. Let these predicates be:

1. There exists a horizontal line of X's
2. There exists a vertical line of X's
3. There exists two X's together somewhere in the map
4. There exists an X in the center spot of the board

Note that the only difference between this hypothesis class and the hypothesis class built on our previous experiment is the absence of the diagonal line of X's predicate. The construction of the experiment went this way so we can still compare the results of the predictor with the previous one but we ensure that we are not counting on the learnable assumption - horizontal or vertical or diagonal - anymore.

More precisely about H_1, assume it is the class of all h's of the form $h = x_1 \wedge x_2 \wedge x_3 \wedge x_4$, where each x_i is a binary digit, indicating h is a binary number from 0000 to 1111 that represents the presence or absence of one of the predicates on the description - $h = 1001$ indicates $h = P(1) \vee P(4)$ for example. Follows by construction that $|H| = 16$.

For the distribution D, we could pick anyone, so we will stick to our former one.

Moreover, for the predictor, we still want to predict whether a state is a win or not for the X player. Assume the same (ϵ, δ) pair as before, so $(\epsilon, \delta) = (0{,}1, 0{,}1)$. According to the uniform convergence sample complexity we will have:

$$m_H(\epsilon, \delta) = m_H(0{,}1, 0{,}1) \leq m_H^{UC}(0{,}05, 0{,}1) \leq \left\lceil \frac{2 \log\left(\frac{2|H|}{0{,}1}\right)}{0{,}1^2} \right\rceil = 1154$$

Since the number of different states in Tic-Tac-Toe is upper bounded by $\sum_{i=1}^{4} \binom{9}{i}\binom{9-i}{i} + \binom{9}{5} = 3264$, if we'd picked an uniform distribution, we would need something in the order of half of the states to make a predictor as good as we are trying to.

In this experiment, our best scenario is $h^* = 1100$ which still makes an error in prediction with probability $\mathcal{L}_{D,g}(h^*) = 0,27$. For our great sample, the one that utilizes $m = 1154$, we get $P\left[\mathcal{L}_{D,g}(h_s) > \mathcal{L}_{D,g}(h^*) + \varepsilon_{0,1}\right] = 0 < 0,1$. In particular, we get $h_s = h^*$ with probability 1. Although it may seems like a good result, it is probably an indication of an upper bound much larger than we would need to get the (ε, δ) we would like. Since we really expect PAC-Learning — and the agnostic version — to make elaborated upper bounds for m, we proceed to our next experiment with $m = 288$.

(*) other realized experiment with $m = 20$. Here we get $P[\mathcal{L}_{D,g}(h_s) > \varepsilon_{0,1}] = 0,087 < 0,1$ which shows that we can get the precision and accuracy desired with a number of samples much lower than the upper bound given by the PAC-Learning $m_H(\varepsilon, \delta)$ function.

For $m = 288$, we got $P[\mathcal{L}_{D,g}(h_s) > \mathcal{L}_{D,g}(h^*) + 0,1] = 0,032 < 0,1$, which guarantees our desired precision and accuracy with a sample size big as a quarter of the calculated upper bound. Now for $m = 72$, we start to get out of bounds with $P[\mathcal{L}_{D,g}(h_s) > \mathcal{L}_{D,g}(h^*) + 0,1] = 0,18 > 0,1$. The significant but not absurd loss of our upper bound argues absolutely what we would expect from agnostic PAC-Learning $m_H^{UC}(\epsilon, \delta)$ function: a much overestimated sample to guarantee the precision and accuracy desired. Our last experiment for this section is $m = 26$. This way we can observe what happens to our agnostic PAC-Learner when we use the sample size upper bound calculated for the classical PAC-Learning experiment. For $m = 26$, hence, we got $P[\mathcal{L}_{D,g}(h_s) > \mathcal{L}_{D,g}(h^*) + 0,1] = 0,31 > 0,1$. With this kind of sample size we don't have guarantees about the precision and accuracy of the learner algorithm and we could not realistically expect better results.