Learning From Noisy Examples (Angluin and Laird, 1987)

Magdalena Markowska
Learnability, Fall 2022

1 Paper contributions

The aim of this paper is propose a learning algorithm capable of dealing with noisy examples.

- learning data: positive examples, some of which are mislabeled by noise before they are exposed to the learning algorithm
- model of noise: the Classification Noise Process
- proposal of a general upper bound on the size of a sample sufficient for learning in finite domains in the presence of noisy data
- evidence that there computationally feasible algorithms for learning in the presence of classification noise in non-trivial domains

2 PAC definition

- purpose: learn a rule that is capable of performing almost correct classification, e.g. recognizing smugglers at the customs
- we need to account for a possibility of being unlucky with the presented data (unlikely)
- number of necessary examples and the computational complexity of the learning procedure are crucial aspects

PAC definition:
Let $L_1, L_2, ...$ be a countable set of subsets of a countable universe U, and D an unknown distribution on the elements of. The task is to identify one such set L_* that correctly classifies the instances into positive and negative examples. Let $EX()$ be a sampling oracle. Each call to the oracle selects an element x ($x \in U$) following D and returns $\langle x, \{0,1\} \rangle$. The success of identifying such set is measured by two parameters, ϵ and δ, which are given as inputs to the identification procedure.

The parameter ϵ is a bound on the difference between the conjectured L_h and the unknown set L_*:

$$d(S,T) = \sum_{x \in S \triangle T} Pr_D(x),$$

where S and T are any subsets of U, $(S \triangle T)$ is the symmetric difference of those subsets ($S \triangle T = (S-T) \cup (T-S)$), and Pr_D is the probability following distribution D. $d(S,T)$ is then the probability that with one call from the oracle we will draw an element that is in either one of S or T sets but not both.

We need δ, a confidence parameter, as a bound on the likelihood of an unlucky event, e.g. exposure to examples not representative of the concept in general. We want to have high confidence that the error is small. The identification procedure is said to do *probably approximately correct identification* if L_* iff:

$$Pr[d(L_*, L_h) \geq \epsilon] \leq \delta,$$

E.g., if we want to have precise classification 85% of the time, with 99% confidence that we are right, we need to set the parameters: $\epsilon = 0.15$ and $\delta = 0.01$.

2.1 Finite classes example

Let $\mathcal{L} = \{L_1, ..., L_N\}$ be any finite set of N rules. $\mathcal{L}$ requests $m = (1/\epsilon)ln(N/\delta)$ examples to return a desired rule. Let L_1 be a rule with error $d(L_1, L_*) \geq \epsilon$. Then $(1 - \epsilon)$ is the likelihood that a random example agrees with L_1, and $(1 - \epsilon)^m$ that m examples agree with L_1. It follows that:

$$(1 - x)^m \leq e^{-mx}$$

Our x is ϵ. So since we have N rules:

$$(1 - \epsilon)^m \leq N(e^{-m\epsilon})$$

We want this to be smaller than δ:

$$N(e^{-m\epsilon}) < \delta$$

If we solve for the requested m above, then the algorithm is PAC-identified.

$$\begin{aligned} N(e^{-m\epsilon}) &< \delta \\ e^{-m\epsilon} &< \delta/N \\ -m\epsilon &< ln(\delta/N) \\ -m\epsilon &< ln(\delta) - ln(N) \\ m\epsilon &> ln(N) - ln(\delta) \\ m\epsilon &> ln(N/\delta) \\ m &> (1/\epsilon)ln(N/\delta) \end{aligned}$$

Now what happens if the algorithm comes across some erroneous examples? Even one such case could prevent finding L_N that belongs to $\mathcal{L}$.

3 How to learn with noisy examples?

First let us look at a model that produces noisy examples.

3.1 The *Classification Noise Process*

- $EX()_0$ is able to draw examples following D without error

- the function determining whether an example is representative of a set is subject to independent random mistakes with some unknown probability $\eta < 1/2$
- procedure: an element x is drawn from U according to D and a coin that comes up with heads with the probability $1 - \eta$
- if we get heads, we mark x correct, if not heads, x will receive the opposite of correct sign
- EX_η is the erroneous oracle
- such probability η occurs independently for each example

If η was equal to $1/2$, the algorithm would not be able to pick up any rules. Another possible problematic case would be if η is very close to $1/2$. In order to make that work, an upper bound η_b is assumed such that $\eta \leq \eta_b \leq 1/2$. While in the case noiseless input, the algorithm runs in polynomial time in $1/\epsilon$ and $1/\delta$, in the case of noisy examples, the polynomial is permitted $1/(1 - 2\eta_b)$ as one of its arguments. Since this quantity is inversely proportional to how close η_b is to $1/2$, the closer η_b is to $1/2$, the longer the algorithm is allowed to run.

Can we do without this upper bound η_b? The answer is yes! Before we show why, let us explore a theorem checking PAC-identification with noisy data.

3.2 How many noisy examples are enough?

In Section 2.1 we showed that:

Theorem 1
If L_i is any hypothesis that agrees with at least

$$m = (1/\epsilon) ln(N/\delta)$$

samples drawn from the $EX_0()$, then

$$Pr[d(L_*, L_i) \geq \epsilon] \leq \delta$$

.

Because the sample size depends on log N, two things follow:

- a large increase of N causes the much smaller growth in the size of sample required
- a small increase in sample size will decrease the confidence limit δ.

How to calculate the sufficient number of noisy examples? Because there is a possibility that there is no L_i that is consistent with all the examples. The way to go around it is to substitute the goal of consistency with the goal of minimizing the number of disagreements with the examples, and permit the number of samples to depend on η_b on the error rate.

Theorem 2: PAC-identification theorem in finite domains with noisy examples
Let $\sigma = \langle x_1, s_1 \rangle, ..., \langle x_n, s_n \rangle$ be a sequence of samples drawn from $EX_\eta()$ oracle, where $x \in U$, and $s = \{+, -\}$. Given if L_i is a possible hypothesis, let $F(L_i, \sigma)$ denote the number of indices j for which L_i disagrees with $\langle x_j, s_j \rangle$, that is, $s_j = +$ and x_j is not in L_i or $s_j = -$ and x_j is in L_i.

If we draw a sequence σ of

$$m \geq \frac{2}{\epsilon^2(1-2\eta_b)} ln\left(\frac{2N}{\delta}\right)$$

samples from an EX_η oracle and find any hypothesis L_i that minimizes $F(L_i, \sigma)$, then

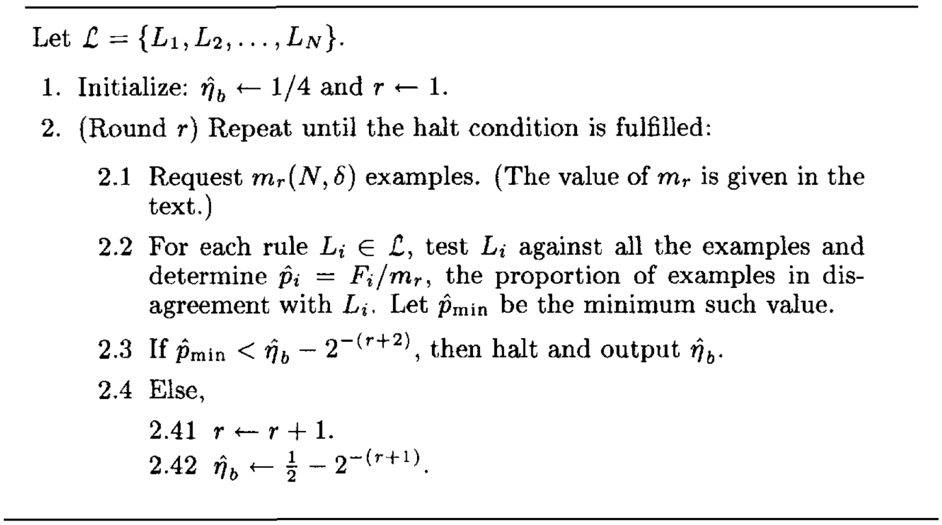
$$Pr[d(L_*, L_i) \geq \epsilon] \leq \delta$$

The number of examples is polynomial in log N, which means that the noise has increased the number of examples we need. Nevertheless this number is still feasible.

3.3 Getting rid of the upper bound for η

- procedure: an iterative search that successively reduces the gap assumed to exist between η and $1/2$
- initial guess: $\eta_b = 1/4$
- then we test by drawing some examples and estimating the failure probability of each of the rules in $\mathcal{L}$
- the smallest empirical failure rate $\hat{p}= F(L_i, \sigma)/m$ is then compared to the current η_b
- if $\hat{p} < \eta_b$ we stop and take $< \eta_b$ as our bound; if $\hat{p} > \eta_b$, we increase the guess for η_b to $3/8, 7/16, 17/32$, etc.
- the size of sample drawn is increased at each iteration

Table 1. The algorithm E.



```
Let $\mathcal{L} = \{L_1, L_2, \ldots, L_N\}$.
1. Initialize: $\hat{\eta}_b \leftarrow 1/4$ and $r \leftarrow 1$.
2. (Round $r$) Repeat until the halt condition is fulfilled:
    2.1 Request $m_r(N, \delta)$ examples. (The value of $m_r$ is given in the text.)
    2.2 For each rule $L_i \in \mathcal{L}$, test $L_i$ against all the examples and determine $\hat{p}_i = F_i/m_r$, the proportion of examples in disagreement with $L_i$. Let $\hat{p}_{\min}$ be the minimum such value.
    2.3 If $\hat{p}_{\min} < \hat{\eta}_b - 2^{-(r+2)}$, then halt and output $\hat{\eta}_b$.
    2.4 Else,
        2.41 $r \leftarrow r + 1$.
        2.42 $\hat{\eta}_b \leftarrow \frac{1}{2} - 2^{-(r+1)}$.
```

Theorem 3
Let

$$m_r(N, \delta) = 2^{2r+3} ln\left(\frac{N2^{r+2}}{\delta}\right).$$

Then with probability greater than $1-\delta$, algorithm E halts on or before round $r^{'} = 1+[log_2(1-2\eta)^{-1}]$ and outputs an estimate $\hat{\eta_b}$ such that $\eta < \hat{\eta_b} < 1/2$

3.4 Minimizing disagreements is computationally expensive

Let n be a positive integer and $PP(n)$ denote the set of all products of a subset of the literals $x_1, ..., x_n$. There are 2^n such products.

- each product π in $PP(n)$ denotes the set of truth-value assignments that satisfy it
- σ will consist of a finite sequence of ordered pairs of the form $\langle a_j, s_j \rangle$, where a_j is a truth-value assignment to the literals, and $s_j \in \{+, -\}$
- If $\pi \in PP(n)$ and σ is a sample sequence, then $F(\pi, \sigma)$ is the number of pairs $\langle a_j, s_j \rangle$ in σ such that $s_j = +$ and $a_j(\pi) = 0$ or $s_j =$ and $a_j(\pi) = 1$. That is, $F(\pi, \sigma)$ is the number of disagreements between π and the sample sequence σ.

Theorem 4
Given positive integers n and c and a sample sequence σ, the problem of determining whether there is an element $\pi \in PP(n)$ such that $F(\pi, \sigma) < c$ is NP-complete.