MACHINE LEARNING

Probably Approximately Correct (PAC) Learning

Alessandro Moschitti

Department of Information Engineering and Computer Science
University of Trento
Email: moschitti@disi.unitn.it
Objectives: defining a well defined statistical framework

- What can we learn and how can we decide if our learning is effective?
- Efficient learning with many parameters
- Trade-off (generalization/and training set error)
- How to represent real world objects
Objectives: defining a well defined statistical framework

- What can we learn and how can we decide if our learning is effective?
- Efficient learning with many parameters
- Trade-off (generalization/and training set error)
- How to represent real world objects
PAC Learning Definition (1)

- Let $c$ be the function (i.e. a concept) we want to learn
- Let $h$ be the learned concept and $x$ an instance (e.g. a person)
- $error(h) = Prob \ [c(x) \neq h(x)]$
- It would be useful if we could find:
  - $Pr(error(h) > \varepsilon) < \delta$
- Given a target error $\varepsilon$, the probability to make a larger error is less $\delta$
This methodology is called Probably Approximately Correct Learning.

The smaller $\varepsilon$ and $\delta$ are the better the learning is.

Problem:
- Given $\varepsilon$ and $\delta$, determine the size $m$ of the training-set.
- Such size may be independent of the learning algorithm.

Let us do it for a simple learning problem.
A simple learning problem

- Learning the concept of **medium-built people** from examples:
  - *Interesting features* are: Height and Weight.
  - The *training-set* of examples has a cardinality of $m$.
    ($m$ people for whom we know if they are medium-built people size, their height and their size).

- Find $m$ to learn this concept **well**.

- The adjective “well” can be expressed with probability error.
Graphical Representation of the target learning problem
Learning Algorithm and Learning Function
Class

1. If no positive examples of the concept are available ⇒ the learned concept is NULL
2. Else the concept is the smallest rectangular (parallel to the axes) containing all positive examples
We don’t consider other complex hypotheses
We don’t consider other complex hypothesis
How good is our algorithm?

- An example $x$ is misclassified if it falls between the two rectangles.
- Let $\varepsilon$ be the measure of the area
- $\Rightarrow$ The error probability (error) of $h$ is $\varepsilon$
  - With which assumption?

[Diagram of two overlapping rectangles with annotations: $h$, $1-\varepsilon$, $\varepsilon$]
Proving PAC Learnability

- Given an error $\varepsilon$ and a probability $\delta$, how many training examples $m$ are needed to learn the concept?
- We can find a bound to $\delta$, i.e. the probability of learning a function $h$ with an error $> \varepsilon$.
- For this purpose, let us compute the probability of selecting a hypothesis $h$ which:
  - correctly classifies $m$ training examples and;
  - shows an error greater than $\varepsilon$.
  - This is a bad function
Probability of Bad Hypotheses

- Given $x$, $P(h(x) = c(x)) < 1 - \varepsilon$
  - since the error of bad function is greater than $\varepsilon$
- Given $\varepsilon$, $m$ examples fall in the rectangle $h$ with a probability $< (1-\varepsilon)^m$
- The probability of choosing a bad hypothesis $h$ is $< (1-\varepsilon)^m \cdot N$
  - where $N$ is the number of hypotheses with an error $> \varepsilon$. 
Upper-bound Computation

- If we set a bound on the probability of bad hypotheses
  \[ N \cdot (1-\varepsilon)^m < \delta \]
- we would be done but we don’t know \( N \)
  \[ \Rightarrow \] we have to find a bound, independent of the number of bad hypothesis.
- Let us divide our rectangle in four strip of area \( \varepsilon/4 \)
Initial Example
A bad hypothesis cannot intersect more than 3 strips at a time.

To intersect 3 edges I can increase the rectangle length but I must decrease the height to have an area \( \leq 1 - \varepsilon \).

Bad hypotheses with error \( > \varepsilon \) are contained in those having an error \( = \varepsilon \).
A bad hypothesis has error $> \varepsilon \Rightarrow$ it has an area $< 1 - \varepsilon$

A rectangle of area $< 1 - \varepsilon$ cannot intersect 4 strips $\Rightarrow$ if the examples fall into all the 4 strips they cannot be part of the same bad hypothesis.

A necessary condition to have a bad hypothesis is that all the $m$ examples are at least outside of one strip.

In other words, when $m$ examples are outside of one of the 4 strips we may have a bad hypothesis.

$\Rightarrow$ the probability of “outside at least one of the strips” $>$ probability of bad hypothesis.
Logic view

- Bad Hypothesis $\Rightarrow$ examples out of at least one strip
  - (viceversa is not true)

- $A \Rightarrow B$
- $P(A) \leq P(B)$
- $P(\text{bad hyp.}) \leq P(\text{out of one strip})$
Upper-bound computation (3)

- \( P(x \text{ out of the target strip}) = (1 - \varepsilon/4) \)
- \( P(m \text{ points out of the target strip}) = (1 - \varepsilon/4)^m \)
- \( P(m \text{ points out of at least one strip}) < 4 \cdot (1 - \varepsilon/4)^m \)

\[ \Rightarrow P(\text{error}(h) > \varepsilon) < 4 \cdot (1 - \varepsilon/4)^m \]
Expliciting $m$

- Our upperbound must be lower than $\delta$, *i.e.*
- $4 \cdot (1 - \varepsilon/4)^m < \delta$
- $\Rightarrow \ln(1 - \varepsilon/4)^m < \delta/4$
- $\Rightarrow m \cdot \ln(1 - \varepsilon/4) < \ln(\delta/4)$
- $\Rightarrow m > \ln(\delta/4) / \ln(1 - \varepsilon/4)$
- change “>” into “<” as $\ln(1 - \varepsilon/4) < 0$
Expliciting $m$

- $-\ln(1-\gamma) = \gamma + \gamma^2/2 + \gamma^3/3 + \ldots$

$\Rightarrow \ln(1-\gamma) = -\gamma - \gamma^2/2 - \gamma^3/3 - \ldots < -\gamma$

$\Rightarrow (1-\gamma) < e^{(-\gamma)}$ it holds strictly for $\gamma > 0$ as in our case

- from $m > \ln(\delta/4)/\ln(1-\varepsilon/4)$

$\Rightarrow m > \ln(\delta/4)/\ln(e^{(-\varepsilon/4)})$

$\Rightarrow m > \ln(\delta/4)/(-\varepsilon/4) \Rightarrow m > \ln(\delta/4) \cdot (4/-\varepsilon)$

$\Rightarrow m > \ln((\delta/4)^{-1}) \cdot (4/\varepsilon) \Rightarrow m > (4/\varepsilon) \cdot \ln(4/\delta)$
### Numeric Examples

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$\delta$</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>148</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>240</td>
</tr>
<tr>
<td>0.1</td>
<td>0.001</td>
<td>332</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$\delta$</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.1</td>
<td>1476</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>2397</td>
</tr>
<tr>
<td>0.01</td>
<td>0.001</td>
<td>3318</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>$\delta$</th>
<th>$m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.1</td>
<td>14756</td>
</tr>
<tr>
<td>0.001</td>
<td>0.01</td>
<td>23966</td>
</tr>
<tr>
<td>0.001</td>
<td>0.001</td>
<td>33176</td>
</tr>
</tbody>
</table>
Formal PAC-Learning Definition

- Let $f$ be the function we want to learn, $f: X\rightarrow I, f \in F$
- $D$ is a probability distribution on $X$
  - used to draw training and test test
- $h \in H$,
  - $h$ is the learned function and $H$ the set of such function class
- $m$ is the training-set size
- $error(h) = \text{Prob } [f(x) \neq h(x)]$
- $F$ is a PAC learnable function class if there is a learning algorithm such that for each $f$, for all distribution $D$ over $X$ and for each $0 < \epsilon, \delta < 1$, produces $h : P(error(h) > \epsilon) < \delta$
Lower Bound on training-set size

Let us reconsider the first bound that we found:

- $h$ is bad: $\text{error}(h) > \epsilon$
- $P(f(x)=h(x))$ for $m$ examples is lower than $(1- \epsilon)^m$
- Multiplying by the number of bad hypotheses we calculate the probability of selecting a bad hypothesis
  - $P(\text{bad hypothesis}) < N \cdot (1- \epsilon)^m < \delta$
  - $P(\text{bad hypothesis}) < N \cdot (e^{-\epsilon})^m = N \cdot e^{-\epsilon m} < \delta$

$\Rightarrow m > (1/\epsilon) \left( \ln(1/\delta) + \ln(N) \right)$

This is a general lower bound
Example

- Suppose we want to learn a boolean function in $n$ variable
- The maximum number of different function are $2^{2^n}$

\[ m > \left( \frac{1}{\varepsilon} \right) \left( \ln\left( \frac{1}{\delta} \right) + \ln\left( 2^{2^n} \right) \right) = \]
\[ = \left( \frac{1}{\varepsilon} \right) \left( \ln\left( \frac{1}{\delta} \right) + 2^n \ln(2) \right) \]
Some Numbers

<table>
<thead>
<tr>
<th>n</th>
<th>epsilon</th>
<th>delta</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.1</td>
<td>245</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.01</td>
<td>268</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>0.1</td>
<td>2450</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>0.01</td>
<td>2680</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>0.1</td>
<td>7123</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>0.01</td>
<td>7146</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>0.1</td>
<td>71230</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>0.01</td>
<td>71460</td>
</tr>
</tbody>
</table>

=------------------------=
References

- PAC-learning:
  - **MY SLIDES**: [http://disi.unitn.it/moschitti/teaching.html](http://disi.unitn.it/moschitti/teaching.html)
  - **MY BOOK**:
    - Artificial Intelligence: a modern approach (Second Edition) by Stuart Russell and Peter Norvig