PAC Learning

Tirgul 2 Part II

November 2016
Waze

• **Task:** suggest\predict the quickest route to a given destination
  • Avoid traffic jams
  • Shortest path
• Predicts **pretty good** routes **most of the time**
  • Pretty good – not always the best, but good enough
  • Most of the time – sometimes fails
Pretty Good = Approximately - \( \epsilon \)
Most of the time = Probably - \( 1 - \delta \)
The output hypothesis is Probably-Approximately Correct
Or in short: PAC
PAC Learning

Probably Approximately Correct
Find a proper hypothesis? How hard can it be?

• Given:
  • Learning algorithm
  • Distribution \( \mathcal{D} \) over \( \mathcal{X} \)
  • Labeling function \( f: \mathcal{X} \rightarrow \{0,1\} \)
  • \( \epsilon \)
  • \( \delta \)

• Goal: find the hypothesis that satisfies \( \epsilon \) and \( \delta \)
  • \( L_D(h) \leq \epsilon \) at least \( 1 - \delta \) of the time

\[ \epsilon = 1, \quad \delta = 1 \]

• We allow it to make horrible mistakes all the time
• How many examples should our learning algorithm be exposed to?
  • 0 – just return any random hypothesis

\[ \epsilon = 0, \quad \delta = 0 \]

• Always Correct!
• How many examples should our learning algorithm be exposed to?
  • Not sure if it’s possible... but if it is, then quite a lot
• $\epsilon$ and $\delta$ determine the amount of examples our algorithms should be exposed to in order to return the proper hypothesis

• In other words:
  • $m_\mathcal{H}: (\epsilon, \delta) \rightarrow \mathbb{N}$
  • $m_\mathcal{H}: (0,1)^2 \rightarrow \mathbb{N}$
PAC Learnability

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_\mathcal{H} : (0,1)^2 \rightarrow \mathbb{N}$ and a \textbf{learning algorithm} with the following property: For every $\epsilon, \delta \in (0,1)$, and for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labeling function $f : \mathcal{X} \rightarrow \{0, 1\}$, if the realizable assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running the learning algorithm on $m \geq m_\mathcal{H}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by $f$, the algorithm returns a hypothesis $h$ such that, with probability of at least $1 - \delta$ (over the choice of the examples),

$$L_{(\mathcal{D},f)}(h) \leq \epsilon$$
Generalizing PAC

• The realizability assumption:
  • $\exists h^* \in \mathcal{H}$ s.t. $\mathbb{P}_{x \sim D}[h^*(x) = f(x)] = 1$
    • In many practical problems, this assumption does not hold!

• Releasing the realizability assumption:
  • Let $D$ be a joint distribution over $\mathcal{X} \times \mathcal{Y}$
True Error Revised

\[ L_D(h) = \mathbb{P}_{(x,y) \sim D}[h(x) \neq y] \]

We wish to find some hypothesis \( h: \mathcal{X} \rightarrow \mathcal{Y} \) that (probably approximately) minimizes the true risk \( L_D(h) \).
Bayes Optimal Classifier
\[ x \in \mathbb{R}^2 \]
\[ y \in \{0, \overline{0}\} \]
\[ h(x) = \begin{cases} 
O & P(O \mid x) > 0.5 \\
O & \text{Otherwise} 
\end{cases} \]

\[ P(O \mid x) > P(O \mid x) \rightarrow O \]

\[ P(O \mid x) < P(O \mid x) \rightarrow O \]
The OPTIMAL predictor - Analysis

• For every probability distribution $\mathcal{D}$ the Bayes Optimal Classifier $h_{\mathcal{D}}$ is optimal, in the sense that no other classifier $g: X \rightarrow \{0,1\}$ has a lower error.

• For every classifier $g$, $L_{\mathcal{D}}(h_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$
The OPTIMAL predictor - proof

• For every hypothesis $g$:

\[
P(g(x) \neq y \mid x) = 1 - P(g(x) = y \mid x)
\]

\[
= 1 - P(g(x) = 1, y = 1 \mid x) - P(g(x) = 0, y = 0 \mid x)
\]

\[
=_{\text{Independence}} 1 - P(g(x) = 1 \mid x) \cdot P(y = 1 \mid x) - P(g(x) = 0 \mid x) \cdot P(y = 0 \mid x)
\]
The OPTIMAL predictor - proof

• Difference between Optimal Bayes Classifier and any other classifier:

\[ P(h_D(x) = y | x) - P(g(x) = y | x) \]

\[ = P(y = 1 | x)[P(h_D(x) = 1 | x) - P(g(x) = 1 | x)] \]

\[ + P(y = 0 | x)[P(h_D(x) = 0 | x) - P(g(x) = 0 | x)] \]

\[ = P(y = 1 | x)[P(h_D(x) = 1 | x) - P(g(x) = 1 | x)] \]

\[ + [1 - P(y = 1 | x)][P(g(x) = 1 | x) - P(h_D(x) = 1 | x)] \]

\[ = [2P(y = 1 | x) - 1][P(h_D(x) = 1 | x) - P(g(x) = 1 | x)] \]
The OPTIMAL predictor - proof

• $[2P(y = 1|x) - 1][P(h_D(x) = 1|x) - P(g(x) = 1|x)]$

• If $P(y = 1|x) > \frac{1}{2} \implies P(h_D(x) = 1|x) = 1$
  $\implies P(h_D(x) = y|x) \geq P(g(x) = y|x)$

• If $P(y = 1|x) < \frac{1}{2} \implies P(h_D(x) = 1|x) = 0$
  $\implies P(h_D(x) = y|x) \geq P(g(x) = y|x)$
  $\implies P(h_D(x) = y|x) \geq P(g(x) = y|x) \implies L_D(h_D) \leq L_D(g)$ \blacksquare
Bayes Optimal Classifier

\[ h(x) = \begin{cases} 
1 & \text{P}(O|x) > 0.5 \\
0 & \text{Otherwise}
\end{cases} \]

But we don’t know \( D \)!!
• Since we don’t know $\mathcal{D}$, we cannot use this optimal predictor $h_\mathcal{D}$.

• We cannot hope that the learning algorithm will find a hypothesis whose error is smaller than the minimal possible error (that of the Bayes predictor)

• ...Instead, we require that the learning algorithm will find a predictor whose error is not much larger than the best possible error of a predictor in some hypothesis class.
Agnostic PAC Learnability

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$, and for every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns a hypothesis $h$ such that, with probability of at least $1 - \delta$ (over the choice of the $m$ training examples),

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$
Agnostic PAC Learnability

• If the realizability assumption holds, agnostic PAC provides the same guarantee as PAC learning.

• Under the definition of Agnostic PAC learning, a learner can still declare success if its error is not much larger than the best error achievable by a predictor from the class $\mathcal{H}$.

  • This is in contrast to PAC learning, in which the learner is required to achieve a small error in absolute terms.
# PAC vs. Agnostic PAC

<table>
<thead>
<tr>
<th></th>
<th>PAC</th>
<th>Agnostic PAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution</td>
<td>$\mathcal{D}$ over $\mathcal{X}$</td>
<td>$\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$</td>
</tr>
<tr>
<td>Truth</td>
<td>$f \in \mathcal{H}$</td>
<td>No realizability assumption</td>
</tr>
<tr>
<td>Risk</td>
<td>$L_{\mathcal{D},f}$</td>
<td>$L_{\mathcal{D}}$</td>
</tr>
<tr>
<td>Training set</td>
<td>$(x_1, \ldots, x_m) \sim \mathcal{D}$</td>
<td>$((x_1, y_1), \ldots, (x_m, y_m)) \sim \mathcal{D}$</td>
</tr>
<tr>
<td></td>
<td>$\forall i, y_i = f(x_i)$</td>
<td></td>
</tr>
<tr>
<td>Goal</td>
<td>$L_{(\mathcal{D},f)}(h) \leq \epsilon$</td>
<td>$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$</td>
</tr>
</tbody>
</table>
Summary

• PAC Learnability
• Releasing the realizability assumption

• Bayes Optimal Predictor
  • We don’t know D...
• Agnostic PAC Learnability