Boost Algorithm

A set of weak learners

Boost

A strong learner
Discrete Adaboost Algorithm

Given: \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x_i \in X, y_i \in Y = \{-1, +1\}\)

Initialise \(D_1(i) = \frac{1}{m}\) \(\rightarrow\) Initial Weight

For \(t = 1, \ldots, T\):

- Find the classifier \(h_t : X \rightarrow \{-1, +1\}\) that minimizes the error with respect to the distribution \(D_t\):
  \[h_t = \arg\min_{h_j \in \mathcal{H}} \varepsilon_j\text{, where } \varepsilon_j = \sum_{i=1}^m D_t(i)[y_i \neq h_j(x_i)]\]

- Prerequisite: \(\varepsilon_t < 0.5\), otherwise stop.

- Choose \(\alpha_t \in \mathbb{R}\), typically \(\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}\) where \(\varepsilon_t\) is the weighted error rate of classifier \(h_t\).

- Update:
  \[D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}\]
  where \(Z_t\) is a normalisation factor (chosen so that \(D_{t+1}\) will be a distribution).

Output the final classifier:

\[H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)\]
## Different Boost Algorithms

Different functions to increase the weight of wrong examples

<table>
<thead>
<tr>
<th>Boost Algorithm</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>$\exp(x)$</td>
</tr>
<tr>
<td>Hinge-loss</td>
<td>linear</td>
</tr>
<tr>
<td>Logitboost</td>
<td>$\ln(1+\exp(-x))$</td>
</tr>
<tr>
<td>0-1 loss</td>
<td>step function</td>
</tr>
<tr>
<td>Brownboost</td>
<td></td>
</tr>
</tbody>
</table>
Problem:
How to choose a set weak learners?
(mainly based on experience and data distribution)

Solution:
Our data should be straightforward, just a set of step-functions.
Each step-function means the value of a variable.

Example:
Weaklearner1:
Do the CT scan (x=1) $\rightarrow$ Not go to Fast Track (y=1)
Not do the CT scan (x=0) $\rightarrow$ go to Fast Track (y=0)

Weaklearner2:
The number of injured parts(x>5) $\rightarrow$ Not go to Fast Track (y=1)
The number of injured parts(x<=5) $\rightarrow$ go to Fast Track (y=0)
Alternating Decision Tree

\[(X_1, X_2, \ldots, X_n)\]

- Do the CT scan
  - yes: +0.3
  - no: -0.3

- The number of injured parts >5
  - yes: +1.5
  - no: -0.2

Problem:
How to find the weight?
Need to use other methods, such as boosting…..

\[Y = \text{sum of all the paths}, \quad \text{sign}(Y) \text{ will be the results}\]