The Dynamics of AdaBoost Weights and applications

B. Caprile, C. Furlanello & S. Merler

ITC-irst – Centro per la Ricerca Scientifica e Tecnologica
I-38050, Povo, Trento
ITALY
The AdaBoost algorithm

- given \( D = \{(x_i, y_i)\}_{i=1}^{N} \), \( x \in X \), \( y \in \{-1, 1\} \)
- initialize \( w_1(i) = 1/N \)
- for \( t = 1, \ldots, T \):
  1. train the base classifier \( M \) using distribution \( w_t(i) \)
  2. get hypothesis \( M_t : X \rightarrow \{-1, +1\} \)
  3. compute model error \( \epsilon_t = \sum_i w_t(i) \Theta[y_i M_t(x_i) = -1] \)
  4. set \( \alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) \)
  5. Update \( w_{t+1}(i) = \frac{1}{Z_t} w_t(i) e^{-\alpha_t y_i M_t(x_i)} \)
- Output the final hypothesis:
  \[
  M(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t M_t(x) \right)
  \]
A toy example
A toy example

The points 1, 2 and 3 are characterized by decreasing margin (with respect to the Bayes classifier)
Evidence on Weights Dynamics

Qualitatively, two behaviours are identified:

– the weight goes (rapidly) to zero;
– the weight goes up and down in a (seemingly) chaotic fashion;
Questions

• (how) can the dynamics of weights be characterized?

• (how) is the dynamics of weights related to the relevance of data points?

• if yes, how can this kind of information be used in practical cases?

• how general are the results that can be obtained in this direction?
Entropy of Weights Distribution

\[ H = - \sum_{i=1}^{L} f_i \log_2 f_i \]
When are these distributions stable?

We tested the hypothesis of same distribution (3000 vs. 5000 AdaBoost steps, maximal tree classifiers, toy example):

p-values from Kolmorogov-Smirnov test
“Easy Points” and “Hard points”

*Easy: low entropy* (Region A)

*Hard: high entropy* (Region B)

# Easy = 289
# Hard = 111

“‘Easy Points’” Are Irrelevant

55 out of 10,000 test points were classified differently by 2 classifiers trained with:
1) all the training data
2) "high entropy" data only

- (stable) cycles can occur \(^a\);
- if AdaBoost cycles, it cycles only among a set of support vectors that achieve the same smallest margin among training examples;
- they give sufficient conditions for AdaBoost to produce a maximum margin classifier when cycling occurs.

\(^a\)in the \(3 \times 3\) case (3 points and 3 weak learners), the weights vector always converge to one of two stable limit cycles
Which kind of distributions are they?

True distribution
Generated Γ distribution

We compared the weights distributions with Γ distributions (K-S test):
the weights distributions can be approximated by Γ distributions.
(partial) Answers

• (how) is the dynamics of weights related to the relevance of data points?
  
  *High Entropy ⇐⇒ Relevant Point (Support Vectors)*

• (how) can the dynamics of weights be characterized?
  
  *Stable distributions (Gamma like), limit cycles*

Applications:

• *Optimal Sampling (Caprile et al., 2002)*

• *Parallelizing Boosting*
Parallel AdaBoost

Given data set $D \equiv \{ (x_i, y_i) \}_{i=1}^N$;

1. Initialize weights $w_i(1) = 1/N$, $i = 1, \ldots, N$;
2. Run AdaBoost for $S$ steps, and store the resulting weights evolution $w_i(s)$, $s = 1, \ldots, S$;
3. For $i = 1, \ldots, N$ estimate distributions $\gamma_i^*$ from weight’s evolutions $w_i(s)$;
4. For $s = S + 1, \ldots, T$: /* This loop can be done in parallel. */
   (a) For $i$ running on the data set generate random weights $w_i^*(s)$ from the corresponding $\gamma_i^*$; normalize weights $w_i^*(s)$ to sum 1;
   (b) train base model $M$ using weights $w_i^*(s)$, obtaining model instance $M_s$;
   (c) compute model error $\epsilon_s$;
   (d) compute model weight $c_s$ according to standard AdaBoost procedure: $c_s = \frac{1}{2} \ln \left( \frac{1-\epsilon_s}{\epsilon_s} \right)$;
5. Output the final model: $H(x) = \sum_{s=1}^T c_s M_s(x)$
P-AdaBoost and margin maximization


- AdaBoost corresponds to the line search minimization of the functional: \[ C(H) = \frac{1}{N} \sum_{i=1}^{N} e^{-y_i H(x_i)} \]

P-AdaBost implements a stochastic minimization strategy.
$\log(C(H))$

$\sin$

$\log(C(H))$

$\text{Gauss}$

$\log(C(H))$

$\text{Gauss2}$

$\log(C(H))$

number of models

number of models

number of models

number of models

$\text{diabetes}$

$\text{thyroid}$

$\text{heart}$

$\text{banana}$
P-Adabbost converges to AdaBoost

Target: Adaboost with 3000 models
P-AdaBoost: results on the sin data set

Adaboost (3000)  Parallel Boosting (100)  Bagging (3000)
<table>
<thead>
<tr>
<th>Data set</th>
<th>met.</th>
<th>3</th>
<th>10</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>diabetes</td>
<td>A</td>
<td>18.4±2.3</td>
<td>14.9±2.3</td>
<td>8.9±1.6</td>
<td>6.2±1.6</td>
<td>4.2±1.5</td>
<td>2.4±1.4</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>8.5±1.4</td>
<td>8.0±1.4</td>
<td>5.4±0.7</td>
<td>4.5±1.2</td>
<td>4.4±1.2</td>
<td>3.0±0.8</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>17.0±2.0</td>
<td>9.0±1.0</td>
<td>8.7±1.5</td>
<td>8.5±1.2</td>
<td>8.6±1.2</td>
<td>8.9±1.2</td>
<td>8.6±1.2</td>
</tr>
<tr>
<td>breast-can.</td>
<td>A</td>
<td>21.1±8.1</td>
<td>16.2±6.4</td>
<td>7.3±3.9</td>
<td>5.7±1.4</td>
<td>2.9±1.5</td>
<td>1.9±1.7</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>6.4±4.4</td>
<td>12.4±8.2</td>
<td>6.5±5.0</td>
<td>5.8±2.9</td>
<td>3.8±2.3</td>
<td>2.7±1.9</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>20.3±13.2</td>
<td>11.7±5.7</td>
<td>7.7±4.8</td>
<td>6.7±4.0</td>
<td>5.4±3.3</td>
<td>5.4±3.5</td>
<td>5.5±2.8</td>
</tr>
<tr>
<td>german</td>
<td>A</td>
<td>25.5±2.0</td>
<td>20.0±2.5</td>
<td>13.1±2.4</td>
<td>10.4±1.9</td>
<td>6.8±1.5</td>
<td>3.6±0.7</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>10.4±2.2</td>
<td>10.4±2.6</td>
<td>8.7±3.3</td>
<td>8.1±1.3</td>
<td>6.5±1.5</td>
<td>4.3±0.8</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>25.1±4.4</td>
<td>14.6±3.8</td>
<td>12.6±1.9</td>
<td>11.3±2.1</td>
<td>12.2±2.4</td>
<td>11.5±2.8</td>
<td>11.6±2.4</td>
</tr>
<tr>
<td>heart</td>
<td>A</td>
<td>18.3±3.5</td>
<td>13.2±3.3</td>
<td>7.0±2.7</td>
<td>3.8±1.6</td>
<td>2.0±1.0</td>
<td>1.2±1.1</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>8.2±3.6</td>
<td>7.2±2.6</td>
<td>6.4±3.3</td>
<td>4.4±2.3</td>
<td>3.6±2.2</td>
<td>2.6±1.3</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>16.7±4.5</td>
<td>10.0±3.2</td>
<td>11.0±3.5</td>
<td>10.3±3.2</td>
<td>10.7±3.1</td>
<td>10.7±3.5</td>
<td>10.4±3.7</td>
</tr>
<tr>
<td>titan</td>
<td>A</td>
<td>13.0±15.9</td>
<td>5.3±4.7</td>
<td>4.6±4.7</td>
<td>1.9±3.9</td>
<td>2.9±3.6</td>
<td>0.8±2.5</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>8.6±9.4</td>
<td>6.3±7.4</td>
<td>6.8±5.8</td>
<td>4.5±5.5</td>
<td>2.9±3.8</td>
<td>1.7±3.2</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>28.0±23.4</td>
<td>14.4±20.6</td>
<td>13.9±22.5</td>
<td>11.2±16.1</td>
<td>7.7±6.9</td>
<td>7.5±6.8</td>
<td>7.6±6.1</td>
</tr>
<tr>
<td>thyroid</td>
<td>A</td>
<td>5.7±3.1</td>
<td>2.9±2.2</td>
<td>1.5±1.0</td>
<td>1.5±2.0</td>
<td>0.4±0.6</td>
<td>0.1±0.4</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>2.5±1.3</td>
<td>2.0±1.8</td>
<td>0.8±1.1</td>
<td>0.8±0.9</td>
<td>0.3±0.6</td>
<td>0.1±0.4</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>5.6±3.4</td>
<td>3.7±2.2</td>
<td>4.5±2.8</td>
<td>3.9±2.7</td>
<td>3.9±2.4</td>
<td>3.7±2.2</td>
<td>3.7±2.4</td>
</tr>
<tr>
<td>banana</td>
<td>A</td>
<td>7.1±1.2</td>
<td>4.1±0.6</td>
<td>1.7±0.4</td>
<td>1.3±0.3</td>
<td>0.9±0.3</td>
<td>0.4±0.1</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>P-A</td>
<td>4.3±0.5</td>
<td>3.0±0.5</td>
<td>1.7±0.5</td>
<td>1.3±0.4</td>
<td>1.2±0.3</td>
<td>0.7±0.2</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>7.6±1.7</td>
<td>4.3±0.9</td>
<td>4.0±0.6</td>
<td>4.0±0.7</td>
<td>4.0±0.7</td>
<td>3.9±0.7</td>
<td>4.0±0.7</td>
</tr>
</tbody>
</table>
An application: parallelizing boosting

- Given $D = \{(x_i, y_i)\}_{i=1}^{N}, x \in X, y \in \{-1, 1\}$, generate and aggregate $B$ models according to the AdaBoost algorithm.
- For $i = 1, \ldots, N$, compute mean $(m_i)$ and variance $(\sigma_i^2)$ of the weight distribution $w_i[1, \ldots, B]$.
- Consider a matrix $A \in \mathcal{M}(N \times C)$ with $C \gg B$.
- For $i = 1, \ldots, N$, $A[i, \ ] \sim \Gamma(C, m_i, \sigma_i^2)$.
- For $j = 1, \ldots, C$, train a model $T_j$ with weights $A[\ , j]$.
- Aggregate the $\{T_j\}_{j=1,\ldots,C}$ according to the AdaBoost algorithm.
An application: parallelizing boosting
(partial) Answers

• (how) can the dynamics of weights be characterized?
  *Limit distribution...Gamma like*

• (how) is the dynamics of weights related to the relevance of data points?
  *High Entropy ⇐⇒ Relevant Point*

• if yes, how can this kind of information be used in practical cases?
  *Parallelizing boosting*

• how general are the results that can be obtained in this direction?
  *We don’t know.*
An Application: Optimal Sampling
An Application: Optimal Sampling
Results (1)
Results (2)
Weights Mean Vs. Variance
Mean Vs. Entropy