The Dynamics of AdaBoost: Cyclic Behavior and Convergence of Margins

Authors: Cynthia Rudin, Ingrid Daubechies, Robert E. Schapire

Speaker: Bernardo Dal Seno
dalseno@elet.polimi.it

6 June 2007
What’s Awaiting You

- AdaBoost
  - The algorithm and some of its features
- How and why AdaBoost works
  - Loss function and margins
- The dynamics of AdaBoost
  - Some examples
  - Asymptotic behavior
- Your questions, any time
Boosting Performance

- PAC (Probably Approximately Correct) theory
  - Some problems admit “strong” learners
  - Some problems admit “weak” learners
  - They are the very same problems!

- Boosting: make weak algorithms strong
  - Several attempt to build boosting algorithms
  - AdaBoost (Freund & Schapire, 1995) is the first “good” boosting algorithm
Boosting Classifiers

- The boosting approach to classification:
  - Devise an algorithm for building weak classifier, i.e., with poor performance (that’s easy!)
  - Obtain a weak classifier for a subset of samples
  - Repeat the previous step T times
  - Combine the classifiers together
- “Weak” means “slightly better than chance”
- Two difficulties:
  - Choice of subsets
  - Combination of classifiers
- AdaBoost addresses these difficulties
Enters AdaBoost

- **Input:**
  - training set: \( (x_i, y_i), \ x_i \in X, \ y_i \in \{-1,+1\}, \ i=1,...,m \)
  - weak classifiers: \( h_j: X \rightarrow \{-1,+1\} \)
  - number of iterations: \( T \)

- **Algorithm:**
  - \( d_1 = \left[ \frac{1}{m} \ldots \frac{1}{m} \right]^T \) (distribution over samples)
  - for \( t=1,...,T \)
    - \( h_t = \arg \min_h \epsilon_t \), \( \epsilon_t = \sum_{h(x_i)y_i=-1} d_{ti} \)
    - \( \alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right) \)
    - \( d_{ti} = \frac{d_{ti}}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases} \)
  - \( Z_t \) normalizing factor

- **Output:**
  - \( H_{\text{final}}(x_i) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x_i) \right) \) (weighted majority)
An Example

- See Schapire’s presentation
Loss Function

- Why AdaBoost works at all?

\[ d_{ti} = \frac{d_{ti}}{Z_t} \exp(-\alpha_t y_i h_t(x_i)) \]

\[ Z_t = \sum_i d_{ti} \exp(-\alpha_t y_i h_t(x_i)) \]

- It happens that minimizes

\[ \prod Z_t = \frac{1}{m} \sum_i \exp(-y_i f(x_i)) \]

where \( f(x_i) = \sum_t \alpha_t h_t(x_i) \)

- It performs a greedy coordinate descent
Boundaries

- Training error is bounded:

  Let $\epsilon_t = 1/2 - \gamma_t$

  $\text{Tr. err.}(H_{\text{final}}) \leq \exp(-2 \sum_t \gamma_t^2)$

- Exponentially fast!

  if $\gamma_t \geq \gamma > 0$

  $\text{Tr. err.}(H_{\text{final}}) \leq e^{-2\gamma^2 T}$
Minimizing Ain’t Enough

- AdaBoost is an exponential-loss minimizer
- But there is more than that:
  - Often, it does not overfit

(by Schapire)
Another Story: Margins

- Margin for sample $i$:

$$
\mu_i = \frac{y_i f(x_i)}{\sum_t |\alpha_t|}
$$

where $f(x_i) = \sum_t \alpha_t h_t(x_i)$

- Margin tells how far we are from uncertainty
- We care mostly about the worst sample

$$
\mu = \min_i \mu_i
$$
Who Cares About Margins?

- High margins are better:
  \[ \text{Generalization error} < \Pr[\mu < \theta] + \tilde{O}\left(\frac{\sqrt{d/m}}{\theta}\right) \]
  with high probability
  - Independent from \( T \), \( d \) is the VC-dimension

- AdaBoost is aggressive
  \[ \Pr[\mu < \theta] \rightarrow 0 \text{ exponentially fast in } T \]
  provided that \( \gamma_t > \theta \)

- Maybe AdaBoost maximizes the margin and hence avoids overfitting
Does AdaBoost Care About Margins?

- Schapire, Freund, Bartlett, and Lee say “Almost” AdaBoost achieves half of the maximum margin.
- Rätsch and Warmuth say it can do even better.

The gap is still wide.
The Matrix Is Everything

- We don’t care about weak classifier answers, just correctness

\[ \epsilon_t = \sum_{h(x_i) y_i = -1} d_{ti} \]

- Suppose we can enumerate weak classifiers, all we need is a classification matrix:

\[ M, \quad M_{ij} = y_i h_j(x_i) \quad (m \times n \text{ matrix}) \]

\[ \epsilon_t = \sum_{M_{ij} = -1} d_{ti} \]

- Final classifier:

\[ H_{\text{final}}(x_i) = \text{sign} \left( \sum_j \lambda_j h_j(x_i) \right) \quad \lambda_j = \frac{\sum_t 1_{h_t = j} \alpha_t}{\sum_t |\alpha_t|} \]
AdaBoost Revisited

- **Input:**
  - training set: \((x_i, y_i), \ x_i \in X, \ y_i \in \{-1, +1\}, \ i = 1, \ldots, m\)
  - classification matrix: \(M, \ M_{ij} = y_i h_j\)
  - number of iterations: \(T\)

- **Algorithm:** \(\lambda_1 = 0\) (weights of classifiers)
  
  for \(t = 1, \ldots, T\)
  
  \[
  d_{ti} = \frac{e^{-(M \lambda_t)_i}}{\sum_j e^{-(M \lambda_t)_j}} \quad \text{(distribution over samples)}
  \]

  \[
  j_t \in \arg\max_j (d_t^T M)_j
  \]

  \[
  r_t = (d_t^T M)_j \quad \text{(edge of classifier } j_t)\]

  \[
  \alpha_t = \frac{1}{2} \ln \left( \frac{1+r_t}{1-r_t} \right)
  \]

  \[
  \lambda_{t+1} = \lambda_t + \alpha_t e_j \quad e_j = [0 \ldots 010 \ldots 0]^T
  \]

- **Output:**
  
  \[
  H_{\text{final}}(x_i) = \text{sign} \left( \sum_{j=1}^n \lambda_{T_j} h_j(x_i) \right) \quad \text{(weighted majority)}
  \]
Compare The “Old” AdaBoost

- **Input:**
  - training set: \( (x_i, y_i), \ x_i \in X, \ y_i \in \{-1,+1\}, \ i=1,...,m \)
  - weak classifiers: \( h_j: X \rightarrow \{-1,+1\} \)
  - number of iterations: \( T \)

- **Algorithm:**

  \[ d_1 = \begin{bmatrix} 1 \\ \vdots \\ 1/m \\ \vdots \\ 1/m \end{bmatrix} \]  
  (distribution over samples)

  for \( t = 1, ..., T \)
  - \( h_t = \arg\min_h \epsilon_t \),  
  - \( \epsilon_t = \sum_{h(x_i) y_i = -1} d_{ti} \)
  - \( \alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right) \)
  - \( d_{ti} = \frac{d_{ti} \times e^{-\alpha_t}}{Z_t} \) if \( y_i = h_t(x_i) \)
  - \( d_{ti} = \frac{d_{ti} \times e^{\alpha_t}}{Z_t} \) if \( y_i \neq h_t(x_i) \)

  \( Z_t \) normalizing factor

- **Output:**

  \( H_{\text{final}}(x_i) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x_i) \right) \)  
  (weighted majority)
Margins, Again

- Redefine the margin:

\[ f(x_i) = \sum_j \bar{\lambda}_j h_j(x_i), \text{ where } \bar{\lambda}_t = \frac{\lambda_t}{\sum_j |\lambda_{tj}|} \]

\[ \mu_i = y_i f(x_i) = y_i \sum_j \bar{\lambda}_j h_j(x_i) = (M \bar{\lambda})_i \]

- Classifier margin:

\[ \mu(\bar{\lambda}) = \min_i (M \bar{\lambda})_i \]

- We want high margin:

\[ \rho = \max_{\bar{\lambda}} \min_i (M \bar{\lambda})_i \text{ (Theoretical margin)} \]

- AdaBoost minimizes:

\[ F(\lambda) = \sum_i e^{-(M \lambda)_i} \]
AdaBoost Dynamics

- In the non-separable case AdaBoost converges

  \[ \text{if } \rho = 0 \quad \lambda_t \to \lambda^* \]

  \[ \lambda^* = \arg\min_{\lambda} \sum_i e^{-M\lambda_i} \]

  - AdaBoost should achieve the maximum margin

- What about the separable case?
  - No unique solution
  - We know nothing about margins

  \[ \text{if } (M\bar{\lambda})_i > 0 \quad \lim_{a \to \infty} F(a\bar{\lambda}) = 0 \]
Yet Another View Of AdaBoost

- AdaBoost in three simple steps (iterated map):
  \[ j_t \in \arg\max_j (d_t^T M)_j \]
  \[ r_t = (d_t^T M)_j \]
  \[ d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij} r_t} \]

- The separable case does not converge to a point:
  - From the min-max theorem:
    \[ \rho = \max_\lambda \min_i (M \lambda)_i = \min_{d} \max_j (d^T M)_i \]
  - Then:
    \[ r_t \geq \rho > 0 \]
    \[ \text{if } d_{ti} > 0 \quad d_{t+1,i} \neq d_{ti} \]
A Detailed Example

\[ M = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \]

\( d \) lies on a simplex

\[ j_t \in \arg\max_j (d_t^T M)_j \]

\[ r_t = (d_t^T M)_{j_t} \]

\[ d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t} \]

Projection of \( d_t \) on the violet triangle

Circles grow with \( t \)
A Detailed Example: Fields

\[ M = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \]

\[ j_t \in \arg\max_j (d_t^T M)_j \]

\[ r_t = (d_t^T M)_{j_t} \]

\[ d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t} \]
A Detailed Example: Maps

\[ M = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \]

\[ j_t \in \arg\max_j (d_t^T M)_j \]

\[ r_t = (d_t^T M)_{j_t} \]

\[ d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t} \]

\[ (d_{t+1}^T M)_{j_t} = 0 \]
A Detailed Examples: Contractions

- There are two cycles of period 3
- Both achieve the maximum margin 1/3
Many Cycles

- Repeated lines
- There exists a stable manifold of 3-cycles
- Weight can be moved around

\[ \begin{pmatrix}
-1 & 1 & 1 \\
-1 & 1 & 1 \\
1 & -1 & 1 \\
1 & 1 & -1 \\
\end{pmatrix} \]

\[ d_t \text{ and } d_t' = d_t + a \text{ behave the same if } \sum_{i \in I_k} a_i = 0 \]

\( I_k \) are groups identical rows

\[
\begin{align*}
 j_t & \in \arg\max_j (d_t^T M)_j \\
r_t & = (d_t^T M)_{j_t} \\
d_{t+1,i} & = \frac{d_{ti}}{1 + M_{ij_t} r_t}
\end{align*}
\]
A Bigger Example

\[ M = \begin{pmatrix} -1 & 1 & \ldots & 1 \\ 1 & -1 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & -1 \end{pmatrix} \]

- There exist (at least) \((m-1)!\) stable cycles
- Those cycles achieve the maximum margin
Support Vectors

- Support vectors are training samples $i$ such that in cycles $d_{ti} > 0$

- If a cycle is stable, for each $i$ either:
  
  $$d_{1,i} = 0$$
  $$\prod_{t=1}^{T_c} (1 + M_{ij_t} r_t) > 1 \quad (d_{ti} \to 0)$$
  $$\prod_{t=1}^{T_c} (1 + M_{ij_t} r_t) = 1 \quad (d_{ti} = d_{t+T_c,i})$$

- The last condition holds for support vectors
  - They are difficult samples
  - AdaBoost concentrates on them

\[
\begin{align*}
  j_t &\in \arg\max_j (d_t^T M)_j \\
  r_t &= (d_t^T M)_{j_t} \\
  d_{t+1,i} &= \frac{d_{ti}}{1 + M_{ij_t} r_t}
\end{align*}
\]
AdaBoost And Margins

- AdaBoost produces the same margin for each support vector and larger margins for other training samples.
- This is the margin of $H_{\text{final}}$.
- AdaBoost does not always converge to the optimal margin.
  - There are counterexamples.
More Than Cycles

- Chaotic-like behavior is also possible
The End

That’s enough for today

Thank you!

And many thanks to C. Rudin and R. Schapire for their work and their material I shamelessly used in this presentation :-)

Thanks also to www.clipartheaven.com for this image
More To Come

- For other exciting seminars, stay tuned on

http://prlt.elet.polimi.it/mediawiki/index.php/Poli_Interest_Group_for_Machine_Learning
References


- Schapire’s lecture on AdaBoost at the Chicago Machine Learning Summer School 2005: [http://videolectures.net/mlss05us_schapire_b/](http://videolectures.net/mlss05us_schapire_b/)