An Online Boosting Algorithm with Theoretical Justifications

Shang-Tse Chen\(^1\)  Hsuan-Tien Lin\(^2\)  Chi-Jen Lu\(^1\)

\(^1\)Academia Sinica  \(^2\)National Taiwan University

June 28, 2012
Motivation

(Batch) Boosting
- weak learners $\rightarrow$ strong learner
- interesting in theory and practice
- need to collect data in advance

Online Learning
- deal with sequential data
- adapt to changing environments
- handle large data sets

Online Weak Learners $\rightarrow$ Online Boosting $\rightarrow$ Online Strong Learner
Brief Review – Batch Boosting

\[(x_1, y_1) \quad (x_2, y_2) \quad (x_T, y_T)\]

training data
Brief Review – Batch Boosting

\[ \begin{align*}
  w_1^{(1)} & \quad w_2^{(1)} & \quad \cdots & \quad w_T^{(1)} \\
  (x_1, y_1) & \quad (x_2, y_2) & \quad \cdots & \quad (x_T, y_T)
\end{align*} \]

training data
Brief Review – Batch Boosting

better than random guessing over $w^{(1)}$

$\alpha^{(1)}$, $h^{(1)}$, $w_1^{(1)}$, $w_2^{(1)}$, $\ldots$, $w_T^{(1)}$

$(x_1, y_1)$, $(x_2, y_2)$, $(x_T, y_T)$

training data
Brief Review – Batch Boosting

\[
\begin{align*}
\alpha^{(1)} & \quad h^{(1)} \\
\begin{array}{cc}
  w_1^{(1)} & w_2^{(1)} \\
  (x_1, y_1) & (x_2, y_2)
\end{array} & \quad \cdots \quad \begin{array}{cc}
  w_1^{(2)} & w_2^{(2)} \\
  (x_T, y_T)
\end{array}
\end{align*}
\]
Brief Review – Batch Boosting

better than random guessing over $w^{(2)}$

$\alpha^{(2)} \ h^{(2)}$

$\alpha^{(1)} \ h^{(1)}$

$w^{(2)}_1 \ w^{(2)}_2 \ \cdots \ \ w^{(2)}_T$

$w^{(1)}_1 \ w^{(1)}_2 \ \cdots \ \ w^{(1)}_T$

$(x_1, y_1) \ \ (x_2, y_2) \ \cdots \ \ (x_T, y_T)$
Brief Review – Batch Boosting

\[
\begin{align*}
\alpha^{(N)} & \quad h^{(N)} \\
& \quad w_1^{(N)} \quad w_2^{(N)} \quad \ldots \quad w_T^{(N)} \\
\vdots & \quad \vdots \\
\alpha^{(2)} & \quad h^{(2)} \\
& \quad w_1^{(2)} \quad w_2^{(2)} \quad \ldots \quad w_T^{(2)} \\
\alpha^{(1)} & \quad h^{(1)} \\
& \quad w_1^{(1)} \quad w_2^{(1)} \quad \ldots \quad w_T^{(1)} \\
(x_1, y_1) & \quad (x_2, y_2) \quad \ldots \quad (x_T, y_T)
\end{align*}
\]
Brief Review – Batch Boosting

\[\alpha^{(N)} \quad h^{(N)}\]

\[w_1^{(N)} \quad w_2^{(N)} \quad \ldots \ldots \quad w_T^{(N)}\]

\[\alpha^{(2)} \quad h^{(2)}\]

\[w_1^{(2)} \quad w_2^{(2)} \quad \ldots \ldots \quad w_T^{(2)}\]

\[\alpha^{(1)} \quad h^{(1)}\]

\[w_1^{(1)} \quad w_2^{(1)} \quad \ldots \ldots \quad w_T^{(1)}\]

Strong Learner

\[(x_1, y_1) \quad (x_2, y_2) \quad \ldots \ldots \quad (x_T, y_T)\]

training data
Brief Review – Online Learning

Input: hypothesis set \( \mathcal{H} \)
randomly choose \( h_1 \in \mathcal{H} \)

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

- receive unlabeled example \( x_t \)
- predict its label with \( h_t(x_t) \)
- receive true label \( y_t \), and suffer loss \( \ell(h_t(x_t), y_t) \)
- update to a new hypothesis \( h_{t+1} \in \mathcal{H} \)

Goal: minimize total loss \( \sum_{t=1}^{T} \ell(h_t(x_t), y_t) \)
At each round $t$:

$\alpha_t^{(N)} \ h^{(N)}_t$

$\alpha_t^{(2)} \ h^{(2)}_t$

$\alpha_t^{(1)} \ h^{(1)}_t$
At each round $t$:
1. receive $x_t$

\[
\begin{align*}
\alpha_t^{(N)} & \quad h_t^{(N)} \\
\alpha_t^{(2)} & \quad h_t^{(2)} \\
\alpha_t^{(1)} & \quad h_t^{(1)}
\end{align*}
\]
Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$

Boosting

$$H_t(x_t) = \text{sign} \left( \sum_{i=1}^{N} \alpha_t^{(i)} h_t^{(i)}(x_t) \right)$$
At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$

$$H_t(x_t) = \text{sign} \left( \sum_{i=1}^{N} \alpha^{(i)}_t h^{(i)}_t(x_t) \right)$$
Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$
At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$

Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$

Online Learning

Boosting

$w_t^{(1)} = 1$
Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$
Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$

Online Learning

Boosting

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$
At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h_t^{(i)}$ and $\alpha_t^{(i)}$
At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h^{(i)}_t$ and $\alpha^{(i)}_t$
Online Boosting Framework

At each round $t$:
1. receive $x_t$
2. predict $H_t(x_t)$
3. receive $y_t$
4. update each $h^{(i)}_t$ and $\alpha^{(i)}_t$

Goal: minimize error rate:

$$\frac{1}{T} \sum_{t=1}^{T} 1[H_t(x_t) \neq y_t]$$
Challenges in Online Boosting

\[
\alpha_t^{(N)} \quad h_t^{(N)}
\]

\[
\alpha_t^{(2)} \quad h_t^{(2)}
\]

\[
\alpha_t^{(1)} \quad h_t^{(1)}
\]

\[
\alpha_{t+1}^{(1)} \quad h_{t+1}^{(1)}
\]

\[
\alpha_{t+1}^{(1)} \quad h_{t+1}^{(1)}
\]
Challenges in Online Boosting

Challenge 1
What's the assumption on the online weak learner?
Challenges in Online Boosting

Challenge 1
What’s the assumption on the online weak learner?

Challenge 2
How to set example weights?
Challenges in Online Boosting

Challenge 1
What’s the assumption on the online weak learner?

Challenge 2
How to set example weights?
1. choose $N$ in the beginning
2. set $\alpha_t$ at each round

Challenge 3
How to combine hypotheses?

1. choose $N$ in the beginning
2. set $\alpha_t$ at each round
Challenge 1 – Online Weak Learning Assumption

Performs better than random guessing under any distributions

Batch

- $w_2=0$

$w_1=1$

- $w_3=0$

Online

+ $w_4=0$

+ $w_5=0$

certain
Challenge 1 – Online Weak Learning Assumption

Performs better than random guessing under any distributions.

**Batch**
- $w_1 = 1$
- $w_2 = 0$
- $w_3 = 0$

**Online**
- $w_4 = 0$
- $w_5 = 0$

Choose $h$ after seeing all examples.

**easy**
Challenge 1 – Online Weak Learning Assumption

Performs better than random guessing under any distributions

Batch

- $w_2 = 0$

- $w_3 = 0$

+ $w_4 = 0$

+ $w_5 = 0$

$w_1 = 1$

Choose $h$ after seeing all examples

Easy

Online

- $w_2 = 0$

- $w_3 = 0$

- $w_4 = 0$

- $w_5 = 0$
Challenge 1 – Online Weak Learning Assumption

Performs better than random guessing under any distributions

**Batch**

- $w_2 = 0$
- $w_3 = 0$
- $w_5 = 0$
- $w_1 = 1$

Choose $h$ after seeing all examples

**Online**

- $w_2 = 0$
- $w_4 = 0$
- $w_3 = 0$
- $w_5 = 0$
- $w_1 = 1$

Choose $h_1$ before seeing any example

**Easy**

**Very Hard**
Key Idea: Sum of example weights large enough
**Challenge 1 – Online Weak Learning Assumption**

- **Key Idea**: Sum of example weights large enough

<table>
<thead>
<tr>
<th>Batch</th>
<th>Online Linear Optimization</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>a fixed hypothesis $h \in \mathcal{H}$</td>
<td>a sequence of hypotheses $h_1, ..., h_T \in \mathcal{H}$</td>
<td></td>
</tr>
</tbody>
</table>

**Advantage**

$$\frac{\sum_{t=1}^{T} w_t y_t h(x_t)}{\sum_{t=1}^{T} w_t} > \gamma$$

$$\frac{\sum_{t=1}^{T} w_t y_t h_t(x_t)}{\sum_{t=1}^{T} w_t} > \gamma - O\left(\sqrt{\frac{1}{\sum_{t=1}^{T} w_t}}\right)$$

**Average Regret**
Challenge 1 – Online Weak Learning Assumption

- **Key Idea:** Sum of example weights large enough

**Batch**
- a fixed hypothesis $h \in \mathcal{H}$

**Online**
- Linear Optimization

**Online**
- a sequence of hypotheses $h_1, \ldots, h_T \in \mathcal{H}$

Advantage:
$$\frac{\sum_{t=1}^{T} w_t y_t h(x_t)}{\sum_{t=1}^{T} w_t} > \gamma$$

New Assumption:
performs better than random guessing when the total weights is larger than $\Omega(1/\gamma^2)$
Challenge 2 – How to set example weights

Weight assignment need to satisfy 2 constraints

(1) sum of weights \( \sum_{t=1}^{T} w_t \) should be larger than \( \Omega(1/\gamma^2) \)

(2) must be set before seeing the remaining examples

- **Adaboost** [Freund & Schapire ’97] doesn’t satisfy condition (1)
  - adjust weights too dramatically
- Scale up or down the weights to meet condition (1)
  - Cannot know the scaling factor without violating condition (2)
- **SmoothBoost** [Servedio ’03] satisfies condition (1), and we prove that it can be successfully adapted to the online setting and meet condition (2)
Challenge 3 – How to combine online weak learners

1. Choose the number of weak learners $N$ in the beginning
   - using **uniform voting weight**, we can achieve error rate $O(\delta)$ by combining at most $O(1/(\delta \gamma^2))$ weak learners
   - we know the existence of a good combination of weak learners

2. Dynamically set voting weights $\alpha_t$ at each round
   - **Online Convex Programming**
     - feasible set: $N$-dimensional probability simplex
     - approaching the performance of a fixed
     \[ \bar{\alpha} = \left( \frac{1}{k}, \ldots, \frac{1}{k}, 0, \ldots, 0 \right), \quad k \leq N \]
   - **Predicting with Expert Advice**
     - $N$ experts, with the $k$-th expert using $\text{sign}(f_t^{(k)})$ to predict $x_t$
     - run the weighted majority algorithm on the $N$ experts
Experiments

Compared Algorithms

1. **Naive Bayes** (weak learner)
2. **Online AdaBoost** [Oza & Russell, ’01]
3. **Online GradientBoost** [Leistner et al., ’09]
4. **Online SmoothBoost** (uniform)

Date Set: **Splice**

Date Set: **Ijcnn1**
Experiments

Comparison of non-uniform voting weight assignment methods

Expert setting is useful for **small** data
- avoid overfitting the small data
- by combining fewer weak learners

OCP is useful for **large** data
- the average regret of OCP diminishes when $T$ is large

### Data Set: Diabetes

### Data Set: WebPage
Conclusion

- We propose a novel online boosting algorithm with nice theoretical properties by solving the 3 main challenges
  - Define appropriate online weak learning assumption
  - Assign example weights
  - Combine online weak learners
- Our algorithm also has superior experimental results on real-world data sets

Thank you. Questions?