Online Boosting Algorithms

Satyen Kale

Yahoo! Labs, NYC
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - (“Attn: Beneficiary Contractor Foreign Money Transfer ...”, spam)
  - (“Let’s meet to discuss QPR –Edo”, not spam)
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - (“Attn: Beneficiary Contractor Foreign Money Transfer ...”, spam)
  - (“Let’s meet to discuss QPR –Edo”, not spam)

- Obtain a classifier by asking a “weak learning algorithm”: 
  - e.g. contains the word “money” \(\Rightarrow\) spam.
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - ("Attn: Beneficiary Contractor Foreign Money Transfer ...", spam)
  - ("Let’s meet to discuss QPR –Edo”, not spam)

- Obtain a classifier by asking a “weak learning algorithm”:
  - e.g. contains the word “money” ⇒ spam.

- Reweight the examples so that “difficult” ones get more attention.
  - e.g. spam that doesn’t contain “money”.
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - ("Attn: Beneficiary Contractor Foreign Money Transfer ...", spam)
  - ("Let’s meet to discuss QPR –Edo", not spam)

- Obtain a classifier by asking a “weak learning algorithm”:
  - e.g. contains the word “money” ⇒ spam.

- Reweight the examples so that “difficult” ones get more attention.
  - e.g. spam that doesn’t contain “money”.

- Obtain another classifier:
  - e.g. empty “to address” ⇒ spam.
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - (“Attn: Beneficiary Contractor Foreign Money Transfer …”, spam)
  - (“Let’s meet to discuss QPR –Edo”, not spam)

- Obtain a classifier by asking a “weak learning algorithm”:
  - e.g. contains the word “money” ⇒ spam.

- Reweight the examples so that “difficult” ones get more attention.
  - e.g. spam that doesn’t contain “money”.

- Obtain another classifier:
  - e.g. empty “to address” ⇒ spam.

- ......
Boosting: An Example

Idea: combine weak “rules of thumb” to form a highly accurate predictor.

Example: email spam detection.

- Given: a set of training examples.
  - (“Attn: Beneficiary Contractor Foreign Money Transfer ...”, spam)
  - (“Let’s meet to discuss QPR –Edo”, not spam)

- Obtain a classifier by asking a “weak learning algorithm”:
  - e.g. contains the word “money” ⇒ spam.

- Reweight the examples so that “difficult” ones get more attention.
  - e.g. spam that doesn’t contain “money”.

- Obtain another classifier:
  - e.g. empty “to address” ⇒ spam.

- ......

- At the end, predict by taking a (weighted) majority vote.
Online Boosting: Motivation

Boosting is well studied in the batch setting, but becomes infeasible when the amount of data is huge.
Online Boosting: Motivation

Boosting is well studied in the batch setting, but becomes infeasible when the amount of data is huge.

Online learning has proven extremely useful:
- one pass of the data, make prediction on the fly.
Online Boosting: Motivation

Boosting is well studied in the batch setting, but becomes infeasible when the amount of data is huge.

Online learning has proven extremely useful:

- one pass of the data, make prediction on the fly.
- works even in an adversarial environment.
  - e.g. spam detection.
Online Boosting: Motivation

Boosting is well studied in the batch setting, but becomes infeasible when the amount of data is huge.

Online learning has proven extremely useful:
- one pass of the data, make prediction on the fly.
- works even in an adversarial environment.
  - e.g. spam detection.

An natural question: **how to extend boosting to the online setting?**
Related Work

Several algorithms exist (Oza and Russell, 2001; Grabner and Bischof, 2006; Liu and Yu, 2007; Grabner et al., 2008).

- mimic offline counterparts.
- achieve great success in many real-world applications.
- no theoretical guarantees.
Related Work

Several algorithms exist (Oza and Russell, 2001; Grabner and Bischof, 2006; Liu and Yu, 2007; Grabner et al., 2008).

- mimic offline counterparts.
- achieve great success in many real-world applications.
- no theoretical guarantees.

Chen et al. (2012): first online boosting algorithms with theoretical guarantees.
- online analogue of weak learning assumption.
- connecting online boosting and smooth batch boosting.
Online Boosting for Classification

Alina Beygelzimer\textsuperscript{1} \quad Satyen Kale\textsuperscript{1} \quad Haipeng Luo\textsuperscript{2}

\textsuperscript{1}Yahoo! Labs, NYC
\textsuperscript{2}Computer Science Department, Princeton University
Batch Boosting

Given a batch of \( T \) examples, \((x_t, y_t) \in \mathcal{X} \times \{-1, 1\}\) for \( t = 1, \ldots, T \). Learner predicts \( \hat{y}_t \in \{-1, 1\}\) for example \( x_t \).
Batch Boosting

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in \{-1, 1\}$ for example $x_t$.

Weak learner (with edge $\gamma$):

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq (\frac{1}{2} - \gamma) T$$
Batch Boosting

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in \{-1, 1\}$ for example $x_t$.

**Weak learner (with edge $\gamma$):**

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq (\frac{1}{2} - \gamma) T$$

**Strong learner (with error rate $\epsilon$):**

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \epsilon T$$
Batch Boosting

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in \{-1, 1\}$ for example $x_t$.

Weak learner (with edge $\gamma$):

$$
\sum_{t=1}^{T} \mathbf{1}\{\hat{y}_t \neq y_t\} \leq \left(\frac{1}{2} - \gamma\right) T
$$

(Schapire, 1990; Freund, 1995)

Strong learner (with error rate $\epsilon$):

$$
\sum_{t=1}^{T} \mathbf{1}\{\hat{y}_t \neq y_t\} \leq \epsilon T
$$
Online Boosting

Given a sequence of $T$ examples $(x_t, y_t) \in X \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner observes $x_t$ and predicts $\hat{y}_t \in \{-1, 1\}$ before seeing $y_t$.

**Weak Online learner** (with edge $\gamma$):

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq (\frac{1}{2} - \gamma) T$$

**Strong Online learner** (with error rate $\epsilon$):

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \epsilon T$$
Online Boosting

Given a sequence of $T$ examples $(x_t, y_t) \in X \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner observes $x_t$ and predicts $\hat{y}_t \in \{-1, 1\}$ before seeing $y_t$.

Weak Online learner (with edge $\gamma$ and excess loss $S$):

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq (\frac{1}{2} - \gamma) T + S$$

Strong Online learner (with error rate $\epsilon$ and excess loss $S'$)

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \epsilon T + S'$$
Online Boosting

Given a sequence of $T$ examples $(x_t, y_t) \in X \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner observes $x_t$ and predicts $\hat{y}_t \in \{-1, 1\}$ before seeing $y_t$.

Weak Online learner (with edge $\gamma$ and excess loss $S$):

$$\sum_{t=1}^{T} 1 \{ \hat{y}_t \neq y_t \} \leq (\frac{1}{2} - \gamma) T + S$$

Our result

Strong Online learner (with error rate $\epsilon$ and excess loss $S'$)

$$\sum_{t=1}^{T} 1 \{ \hat{y}_t \neq y_t \} \leq \epsilon T + S'$$
Online Boosting

Given a sequence of $T$ examples $(x_t, y_t) \in X \times \{-1, 1\}$ for $t = 1, \ldots, T$. Learner observes $x_t$ and predicts $\hat{y}_t \in \{-1, 1\}$ before seeing $y_t$.

**Weak Online learner** (with edge $\gamma$ and excess loss $S$):

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq (\frac{1}{2} - \gamma) T + S
\]

↓ Our result

**Strong Online learner** (with error rate $\epsilon$ and excess loss $S'$)

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \epsilon T + S'
\]

this talk: $S = \frac{1}{\gamma}$ (corresponds to $\sqrt{T}$ regret)
Main Results

Parameters of interest:

\( N = \) number of weak learners (of edge \( \gamma \)) needed to achieve error rate \( \epsilon \).

\( T_\epsilon = \) minimal number of examples s.t. error rate is \( \epsilon \).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>( N )</th>
<th>( T_\epsilon )</th>
<th>Optimal?</th>
<th>Adaptive?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online BBM</td>
<td>( O\left(\frac{1}{\gamma^2 \ln \frac{1}{\epsilon}}\right) )</td>
<td>( \tilde{O}\left(\frac{1}{\epsilon \gamma^2}\right) )</td>
<td>( \checkmark )</td>
<td>( \times )</td>
</tr>
<tr>
<td>AdaBoost.OL</td>
<td>( O\left(\frac{1}{\epsilon \gamma^2}\right) )</td>
<td>( \tilde{O}\left(\frac{1}{\epsilon^2 \gamma^4}\right) )</td>
<td>( \times )</td>
<td>( \checkmark )</td>
</tr>
<tr>
<td>Chen et al. (2012)</td>
<td>( O\left(\frac{1}{\epsilon \gamma^2}\right) )</td>
<td>( \tilde{O}\left(\frac{1}{\epsilon \gamma^2}\right) )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
</tbody>
</table>
Structure of Online Boosting

\[ x_1 \]

\[ \text{Booster} \]
Structure of Online Boosting

\[
\begin{align*}
WL_1 & \quad \text{predict} \quad \hat{y}_1^1 \\
WL_2 & \quad \text{predict} \quad \hat{y}_2^2 \\
\ldots & \\
WL_N & \quad \text{predict} \quad \hat{y}_N^N
\end{align*}
\]
Structure of Online Boosting

\[ \hat{y}_1 = W_L^1(x_1) \]

\[ \hat{y}_2 = W_L^2(x_1) \]

\[ \hat{y}_N = W_L^N(x_1) \]

\[ x_1, \hat{y}_1 \]

\[ x_1, \hat{y}_2 \]

\[ x_1, \hat{y}_N \]
Structure of Online Boosting

\[ WL^1 \]
predict

\[ WL^2 \]
predict

... 

\[ WL^N \]
predict

\( x_1 \quad \hat{y}_1 \quad y_1 \)

Update w.p. \( p_1 \) on \((x_1, y_1)\)

Update w.p. \( p_2 \) on \((x_1, y_1)\)

...
Structure of Online Boosting

\[ \hat{y}_1 = WL_1(x_1, y_1) \]

\[ \hat{y}_2 = WL_2(x_1, y_1) \]

... 

\[ \hat{y}_N = WL_N(x_1, y_1) \]

w.p. \( p_1 \)

(\( x_1, y_1 \)
Structure of Online Boosting
Structure of Online Boosting

\[ \hat{y}_2^1 = WL_1 \text{ predict} \]
\[ \hat{y}_2^2 = WL_2 \text{ predict} \]
\[ \ldots \]
\[ \hat{y}_2^N = WL_N \text{ predict} \]

\[ x_2 \]

\[ w_1 \text{ update w.p. } p_1 (x_2, y_2) \]
\[ w_2 \text{ update w.p. } p_2 (x_2, y_2) \]
\[ \ldots \]
\[ w_N \text{ update w.p. } p_N (x_2, y_2) \]
Structure of Online Boosting

\[ WL_1^{\text{predict}} \]
\[ WL_2^{\text{predict}} \]
\[ \ldots \]
\[ WL_N^{\text{predict}} \]

\[ x_2 \]
\[ \hat{y}_2 \]
\[ x_2 \]
\[ \hat{y}_2 \]
\[ x_2 \]
\[ \hat{y}_2 \]
\[ x_2 \]
\[ \hat{y}_2 \]
Structure of Online Boosting

\[ WL_1 \] predict
\[ x_2, \hat{y}_2, y_2 \]

\[ WL_2 \] predict
\[ x_2, \hat{y}_2^1 \]
\[ (x_2, y_2) \]

\[ WL_1 \] update
\[ w.p. \ p_2^1 \]

\[ WL_1 \]
\[ WL_2 \]

\[ WL_N \] predict
\[ x_2, \hat{y}_2^N \]
\[ (x_2, y_2) \]

\[ WL_N \] update
\[ w.p. \ p_2^N \]
Structure of Online Boosting

\[ W_{L1} \text{ predict} \]
\[ W_{L2} \text{ predict} \]
\[ \ldots \]
\[ W_{LN} \text{ predict} \]

\[ x_t, \hat{y}_t, y_t \]

\[ W_{L1} \text{ update} \]
\[ W_{L2} \text{ update} \]
\[ \ldots \]
\[ W_{LN} \text{ update} \]

\[ \text{w.p. } p_t^1 (x_t, y_t) \]
\[ \text{w.p. } p_t^2 (x_t, y_t) \]
\[ \text{w.p. } p_t^N (x_t, y_t) \]
Boosting as a Drifting Game (Schapire, 2001; Luo and Schapire, 2014)

Batch boosting can be analyzed using drifting games.
Batch boosting can be analyzed using drifting games.

**Online version:** sequence of potentials $\Phi_i(s)$ s.t.

\[
\begin{align*}
\Phi_N(s) & \geq 1 \{ s \leq 0 \}, \\
\Phi_{i-1}(s) & \geq \left( \frac{1}{2} - \frac{\gamma}{2} \right) \Phi_i(s - 1) + \left( \frac{1}{2} + \frac{\gamma}{2} \right) \Phi_i(s + 1).
\end{align*}
\]
Batch boosting can be analyzed using drifting games.

**Online version**: sequence of potentials $\Phi_i(s)$ s.t.

\[
\begin{align*}
\Phi_N(s) &\geq 1\{s \leq 0\}, \\
\Phi_{i-1}(s) &\geq \left(\frac{1}{2} - \frac{\gamma}{2}\right)\Phi_i(s - 1) + \left(\frac{1}{2} + \frac{\gamma}{2}\right)\Phi_i(s + 1).
\end{align*}
\]

Online boosting algorithm using $\Phi_i$:

- **prediction**: majority vote.
Boosting as a Drifting Game
(Schapire, 2001; Luo and Schapire, 2014)

Batch boosting can be analyzed using drifting games.

**Online version**: sequence of potentials \( \Phi_i(s) \) s.t.

\[
\begin{align*}
\Phi_N(s) &\geq 1 \{ s \leq 0 \}, \\
\Phi_{i-1}(s) &\geq \left( \frac{1}{2} - \frac{\gamma}{2} \right) \Phi_i(s - 1) + \left( \frac{1}{2} + \frac{\gamma}{2} \right) \Phi_i(s + 1).
\end{align*}
\]

Online boosting algorithm using \( \Phi_i \):

- **prediction**: majority vote.
- **update**: \( p^i_t = \Pr[(x_t, y_t) \text{ sent to } WL^i] \propto w^i_t \) where \( w^i_t = \text{difference in potentials if example is misclassified or not} \).
Mistake Bound

Generalized drifting games analysis implies

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_\infty.$$
Mistake Bound

Generalized drifting games analysis implies

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_\infty. 
\]

So we want small \(\|w^i\|_\infty\).

- exponential potential (corresponding to AdaBoost) does not work.
Mistake Bound

Generalized drifting games analysis implies

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_{\infty} \leq \epsilon\]

So we want small \(\|w^i\|_{\infty}\).

- exponential potential (corresponding to AdaBoost) does not work.
- Boost-by-Majority (Freund, 1995) potential works well!
Mistake Bound

Generalized drifting games analysis implies

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_\infty. 
\]

So we want small \(\|w^i\|_\infty\).

- exponential potential (corresponding to AdaBoost) does not work.
- Boost-by-Majority (Freund, 1995) potential works well!
  - \(w^i_t = \text{Pr}[k_t^i \text{ heads in } N - i \text{ flips of a } \frac{\gamma}{2}\text{-biased coin}]\)
Mistake Bound

Generalized drifting games analysis implies

\[
\sum_{t=1}^{T} \mathbf{1}\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_{\infty} \leq \epsilon \quad \overset{\text{=} S'}{=} S'.
\]

So we want small \(\|w^i\|_{\infty}\).

- exponential potential (corresponding to AdaBoost) does not work.
- Boost-by-Majority (Freund, 1995) potential works well!
  - \(w^i_t = \Pr[k^i_t \text{ heads in } N - i \text{ flips of a } \frac{\gamma}{2}\text{-biased coin}] \leq \frac{4}{\sqrt{N-i}}\)
Mistake Bound

Generalized drifting games analysis implies

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \Phi_0(0) T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_\infty \leq \epsilon \leq T + (S + \frac{1}{\gamma}) \sum_i \|w^i\|_\infty = S'.$$

So we want small $\|w^i\|_\infty$.

- exponential potential (corresponding to AdaBoost) does not work.
- Boost-by-Majority (Freund, 1995) potential works well!
  - $w_t^i = \Pr[k_t^i \text{ heads in } N - i \text{ flips of a } \gamma/2\text{-biased coin}] \leq \frac{4}{\sqrt{N-i}}$

Online BBM: to get $\epsilon$ error rate, needs $N = O\left(\frac{1}{\gamma^2 \ln\left(\frac{1}{\epsilon}\right)}\right)$ weak learners and $T_\epsilon = O\left(\frac{1}{\epsilon \gamma^2}\right)$ examples. (Optimal)
The draw back of BBM (or Chen et al. (2012)) is the lack of adaptivity.

- requires \( \gamma \) as a parameter.
Drawback of Online BBM

The draw back of BBM (or Chen et al. (2012)) is the lack of adaptivity.

- requires $\gamma$ as a parameter.
- treats each weak learner equally: predicts via simple majority vote.
Drawback of Online BBM

The draw back of BBM (or Chen et al. (2012)) is the lack of adaptivity.

- requires $\gamma$ as a parameter.
- treats each weak learner equally: predicts via simple majority vote.

Adaptivity is the key advantage of AdaBoost!

- different weak learners weighted differently based on their performance.
Batch boosting finds a combination of weak learners to minimize some loss function using coordinate descent. (Breiman, 1999)
Adaptivity via Online Loss Minimization

Batch boosting finds a combination of weak learners to minimize some loss function using coordinate descent. (Breiman, 1999)

- AdaBoost: exponential loss
- AdaBoost.L: logistic loss
Adaptivity via Online Loss Minimization

Batch boosting finds a combination of weak learners to minimize some loss function using coordinate descent. (Breiman, 1999)

- **AdaBoost**: exponential loss
- **AdaBoost.L**: logistic loss

We generalize it to the online setting:

- replace line search with online gradient descent.
Adaptivity via Online Loss Minimization

Batch boosting finds a combination of weak learners to minimize some loss function using coordinate descent. (Breiman, 1999)

- **AdaBoost**: exponential loss
- **AdaBoost.L**: logistic loss

We generalize it to the online setting:

- replace line search with online gradient descent.
- exponential loss does not work again, use logistic loss to get adaptive online boosting algorithm **AdaBoost.OL**.
Mistake Bound

If WL\(i\) has edge \(\gamma_i\), then

\[
\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)
\]
Mistake Bound

If $WL^i$ has edge $\gamma_i$, then

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)$$

Suppose $\gamma_i \geq \gamma$, then to get $\epsilon$ error rate AdaBoost.OL needs $N = O\left(\frac{1}{\epsilon \gamma^2}\right)$ weak learners and $T_\epsilon = O\left(\frac{1}{\epsilon^2 \gamma^4}\right)$ examples.
Mistake Bound

If $WL^i$ has edge $\gamma_i$, then

$$\sum_{t=1}^{T} 1\{\hat{y}_t \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)$$

Suppose $\gamma_i \geq \gamma$, then to get $\epsilon$ error rate AdaBoost.OL needs $N = \mathcal{O}\left(\frac{1}{\epsilon^2 \gamma^2}\right)$ weak learners and $T_\epsilon = \mathcal{O}\left(\frac{1}{\epsilon^2 \gamma^4}\right)$ examples.

Not optimal but adaptive.
Results

Available in Vowpal Wabbit 8.0.

- command line option: \texttt{--boosting}.
- VW as the default “weak” learner (a rather strong one!)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>VW baseline</th>
<th>Online BBM</th>
<th>AdaBoost.OL</th>
<th>Chen et al. 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>20news</td>
<td>0.0812</td>
<td>0.0775</td>
<td>0.0777</td>
<td>0.0791</td>
</tr>
<tr>
<td>a9a</td>
<td>0.1509</td>
<td>0.1495</td>
<td>0.1497</td>
<td>0.1509</td>
</tr>
<tr>
<td>activity</td>
<td>0.0133</td>
<td>0.0114</td>
<td>0.0128</td>
<td>0.0130</td>
</tr>
<tr>
<td>adult</td>
<td>0.1543</td>
<td>0.1526</td>
<td>0.1536</td>
<td>0.1539</td>
</tr>
<tr>
<td>bio</td>
<td>0.0035</td>
<td>0.0031</td>
<td>0.0032</td>
<td>0.0033</td>
</tr>
<tr>
<td>census</td>
<td>0.0471</td>
<td>0.0469</td>
<td>0.0469</td>
<td>0.0469</td>
</tr>
<tr>
<td>covtype</td>
<td>0.2563</td>
<td>0.2347</td>
<td>0.2495</td>
<td>0.2470</td>
</tr>
<tr>
<td>letter</td>
<td>0.2295</td>
<td>0.1923</td>
<td>0.2078</td>
<td>0.2148</td>
</tr>
<tr>
<td>maptaskcoref</td>
<td>0.1091</td>
<td>0.1077</td>
<td>0.1083</td>
<td>0.1093</td>
</tr>
<tr>
<td>nomao</td>
<td>0.0641</td>
<td>0.0627</td>
<td>0.0635</td>
<td>0.0627</td>
</tr>
<tr>
<td>poker</td>
<td>0.4555</td>
<td>0.4312</td>
<td>0.4555</td>
<td>0.4555</td>
</tr>
<tr>
<td>rcv1</td>
<td>0.0487</td>
<td>0.0485</td>
<td>0.0484</td>
<td>0.0488</td>
</tr>
<tr>
<td>vehv2binary</td>
<td>0.0292</td>
<td>0.0286</td>
<td>0.0291</td>
<td>0.0284</td>
</tr>
</tbody>
</table>
Online Boosting for Regression

Alina Beygelzimer\(^1\)  Elad Hazan\(^2\)  Satyen Kale\(^1\)  Haipeng Luo\(^2\)

\(^1\)Yahoo! Labs, NYC

\(^2\)Computer Science Department, Princeton University
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \to [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \rightarrow [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

**Boosting** \(\equiv\) greedy stagewise algorithm for fitting of additive models.
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \rightarrow [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

**Boosting** ≡ greedy stagewise algorithm for fitting of additive models.

I.e. given alg to fit model in \(\mathcal{F}\), fit an additive model in \(\text{span}(\mathcal{F})\)
Regression Setting

Setup:
- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \rightarrow [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

Boosting \(\equiv\) greedy stagewise algorithm for fitting of additive models.

I.e. given alg to fit model in \(\mathcal{F}\), fit an additive model in \(\text{span}(\mathcal{F})\)

Typically, by “greedily fitting the residual,” as in Basis Pursuit.
Regression Setting

Setup:

- Examples $(x, y) \in \mathcal{X} \times [-1, 1]
- Loss of predicting $\hat{y}$ for $(x, y)$ is $(y - \hat{y})^2$
- $\mathcal{F}$ is a base class of regressors $f : \mathcal{X} \rightarrow [-1, 1]$
- $\text{span}(\mathcal{F})$ is set of linear combinations of regressors in $\mathcal{F}$

Boosting $\equiv$ greedy stagewise algorithm for fitting of additive models.
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \to [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

Boosting \(\equiv\) greedy stagewise algorithm for fitting of additive models.

- **Input:** a batch \(S\) of examples, number of boosting steps \(N\), and step size parameter \(\eta\).
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \rightarrow [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

Boosting \equiv \) greedy stagewise algorithm for fitting of additive models.

- **Input:** a batch \(S\) of examples, number of boosting steps \(N\), and step size parameter \(\eta\).
- Set \(g\) to be the constant 0 model.
Regression Setting

Setup:
- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(F\) is a base class of regressors \(f : \mathcal{X} \to [-1, 1]\)
- \(\text{span}(F)\) is set of linear combinations of regressors in \(F\)

Boosting \(\equiv\) greedy stagewise algorithm for fitting of additive models.

- **Input:** a batch \(S\) of examples, number of boosting steps \(N\), and step size parameter \(\eta\).
- Set \(g\) to be the constant 0 model.
- Repeat for \(N\) steps, starting with 0: find
  
  \[
  f = \arg\min_{f \in F} \sum_{(x, y) \in S} \left( y - g(x) - \eta f(x) \right)^2
  \]
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \rightarrow [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

Boosting ≡ greedy stagewise algorithm for fitting of additive models.

- **Input:** a batch \(S\) of examples, number of boosting steps \(N\), and step size parameter \(\eta\).
- Set \(g\) to be the constant 0 model.
- Repeat for \(N\) steps, starting with 0: find

\[
\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{(x, y) \in S} \left(y - g(x) - \eta f(x)\right)^2
\]
Regression Setting

Setup:

- Examples \((x, y) \in \mathcal{X} \times [-1, 1]\)
- Loss of predicting \(\hat{y}\) for \((x, y)\) is \((y - \hat{y})^2\)
- \(\mathcal{F}\) is a base class of regressors \(f : \mathcal{X} \to [-1, 1]\)
- \(\text{span}(\mathcal{F})\) is set of linear combinations of regressors in \(\mathcal{F}\)

Boosting \(\equiv\) greedy stagewise algorithm for fitting of additive models.

- **Input**: a batch \(S\) of examples, number of boosting steps \(N\), and step size parameter \(\eta\).
- Set \(g\) to be the constant 0 model.
- Repeat for \(N\) steps, starting with 0: find

\[
\begin{align*}
    f &= \arg \min_{f \in \mathcal{F}} \sum_{(x,y) \in S} \left( (y - g(x) - \eta f(x))^2 \right) \\
    \text{and update} \quad g &\leftarrow g + \eta f.
\end{align*}
\]
Batch Boosting: Convergence

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$. 

Weak learner (a.k.a. ERM):

$$\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in F} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2 \quad (\text{Friedman, 2001; Mason et al., 2000})$$

Strong learner:

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \inf_{f \in \text{span}(F)} \sum_{t=1}^{T} (y_t - f(x_t))^2 \quad \Delta f \to 0 \text{ as } N \to \infty.$$
Batch Boosting: Convergence

Given a *batch* of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$.

**Weak learner** (a.k.a. ERM):

$$
\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2
$$
Batch Boosting: Convergence

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$.

**Weak learner** (a.k.a. ERM):

$$\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in F} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2$$

**Strong learner**:

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \inf_{f \in \text{span}(\mathcal{F})} \sum_{t=1}^{T} (y_t - f(x_t))^2$$
Batch Boosting: Convergence

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$.

**Weak learner** (a.k.a. ERM):

$$\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2$$

**Strong learner**: For any $f \in \text{span}(\mathcal{F})$,

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T} (y_t - f(x_t))^2 + \Delta_f$$

$\Delta_f \to 0$ as $N \to \infty$. 

(Friedman, 2001; Mason et al., 2000; Zhang and Yu, 2005)
Batch Boosting: Convergence

Given a batch of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$.

**Weak learner** (a.k.a. ERM):

$$
\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2
$$

(Friedman, 2001; Mason et al., 2000)

(Zhang and Yu, 2005)

**Strong learner**: For any $f \in \text{span} (\mathcal{F})$,

$$
\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T} (y_t - f(x_t))^2 + \Delta_f
$$

$\Delta_f \to 0$ as $N \to \infty$. 
Online Boosting

Given a sequence of \( T \) examples, \((x_t, y_t) \in \mathcal{X} \times [-1, 1]\) for \( t = 1, \ldots, T \). Learner predicts \( \hat{y}_t \in [-1, 1] \) for example \( x_t \) before observing \( y_t \).

**Weak online learner** (a.k.a. online ERM):

\[
\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2 + R(T)
\]

**Strong online learner**: For any \( f \in \text{span}(\mathcal{F}) \),

\[
\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T} (y_t - f(x_t))^2 + R'_f(T)
\]

\( R_f(T) \to 0 \) as \( N \to \infty \).
Online Boosting

Given a sequence of $T$ examples, $(x_t, y_t) \in \mathcal{X} \times [-1, 1]$ for $t = 1, \ldots, T$. Learner predicts $\hat{y}_t \in [-1, 1]$ for example $x_t$ before observing $y_t$.

**Weak online learner** (a.k.a. *online ERM*):

$$\sum_{t=1}^{T} (y_t - \eta \hat{y}_t)^2 \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} (y_t - \eta f(x_t))^2 + R(T)$$

↓ Our result

**Strong online learner**: For any $f \in \text{span}(\mathcal{F})$,

$$\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \sum_{t=1}^{T} (y_t - f(x_t))^2 + R'_f(T)$$

$R_f(T) \rightarrow 0$ as $N \rightarrow \infty$. 
Structure of Batch Boosting

**Input:** batch of examples \( \{(x_t, y_t) \mid t = 1, 2, \ldots, T\} \), step-size \( \eta \)
Structure of Batch Boosting

**Input:** batch of examples \( \{(x_t, y_t) \mid t = 1, 2, \ldots, T\} \), step-size \( \eta \)

Set “pseudo-labels” \( \tilde{y}^1_t = y_t \) for all \( t \)
Structure of Batch Boosting

**Input:** batch of examples \( \{(x_t, y_t) \mid t = 1, 2, \ldots, T \} \), step-size \( \eta \)

Set “pseudo-labels” \( \tilde{y}_t^1 = y_t \) for all \( t \)

For \( i = 1, 2, \ldots, N \)

1. Train weak learner on examples \( \{(x_t, \tilde{y}_t^i) \mid t = 1, 2, \ldots T \} \) with step-size \( \eta \)
Structure of Batch Boosting

**Input:** batch of examples \( \{ (x_t, y_t) \mid t = 1, 2, \ldots, T \} \), step-size \( \eta \)

Set “pseudo-labels” \( \tilde{y}^1_t = y_t \) for all \( t \)

For \( i = 1, 2, \ldots, N \)

1. Train weak learner on examples \( \{ (x_t, \tilde{y}^i_t) \mid t = 1, 2, \ldots T \} \) with step-size \( \eta \)
2. Obtain predictions \( \hat{y}_t^i \) for all \( t \)
Structure of Batch Boosting

**Input:** batch of examples \( \{(x_t, y_t) \mid t = 1, 2, \ldots, T\} \), step-size \( \eta \)

Set “pseudo-labels” \( \tilde{y}_t^1 = y_t \) for all \( t \)

For \( i = 1, 2, \ldots, N \)

1. Train weak learner on examples \( \{(x_t, \tilde{y}_t^i) \mid t = 1, 2, \ldots, T\} \) with step-size \( \eta \)
2. Obtain predictions \( \hat{y}_t^i \) for all \( t \)
3. Compute pseudo-labels \( \tilde{y}_t^{i+1} = \tilde{y}_t^i - \eta \hat{y}_t^i \) for \( t \)
Structure of Online Boosting for Regression

\[ x_1 \]

Booster
Structure of Online Boosting for Regression

\[ \mathbb{W}L_1 \text{ predict} \]
\[ \hat{y}_1 \]

\[ \mathbb{W}L_2 \text{ predict} \]
\[ \hat{y}_2 \]

\[ \ldots \]

\[ \mathbb{W}L_N \text{ predict} \]
\[ \hat{y}_N \]

\[ x_1 \rightarrow \mathbb{W}L_1 \rightarrow \hat{y}_1 \rightarrow x_1 \]

\[ x_1 \rightarrow \mathbb{W}L_2 \rightarrow \hat{y}_2 \rightarrow \hat{y}_1 \rightarrow x_1 \]

\[ x_1 \rightarrow \mathbb{W}L_N \rightarrow \hat{y}_N \rightarrow \hat{y}_1 \rightarrow x_1 \]
Structure of Online Boosting for Regression

\[
\begin{align*}
WL_1 & \text{ predict } \hat{y}_1^1 \\
WL_2 & \text{ predict } \hat{y}_1^2 \\
\ldots \\
WL_N & \text{ predict } \hat{y}_1^N \\
\end{align*}
\]

\[
\begin{align*}
WL_1 & \text{ update } (x_1, \tilde{y}_1^1) \\
WL_2 & \text{ update } (x_1, \tilde{y}_1^2) \\
\ldots \\
WL_N & \text{ update } (x_1, \tilde{y}_1^N) \\
\end{align*}
\]
Structure of Online Boosting for Regression

\[
\begin{align*}
&WL_1^1 \quad \text{predict} \quad x_1 \quad \hat{y}_1^1 \quad \hat{y}_1 \quad y_1 \\
&WL_2^2 \quad \text{predict} \quad x_1 \quad \hat{y}_1^2 \quad \hat{y}_1 \\
&\quad \vdots \\
&WL_N^N \quad \text{predict} \quad x_1 \quad \hat{y}_1^N \quad \hat{y}_1
\end{align*}
\]
Structure of Online Boosting for Regression

\[ \hat{y}_1 = WL_1^{\text{predict}}(x_1) \]

\[ (x_1, \hat{y}_1) = WL_1^{\text{update}}(x_1, \tilde{y}_1) \]

\[ \hat{y}_2 = WL_2^{\text{predict}}(x_1) \]

\[ (x_1, \hat{y}_2) = WL_2^{\text{update}}(x_1, \tilde{y}_1) \]

\[ \ldots \]

\[ \hat{y}_N = WL_N^{\text{predict}}(x_1) \]

\[ (x_1, \hat{y}_N) = WL_N^{\text{update}}(x_1, \tilde{y}_1) \]
Structure of Online Boosting for Regression

\[
\hat{y}_1 = \text{WL}\cdot x_2 \\
\hat{y}_2 = \text{WL}\cdot x_2 \\
\vdots \\
\hat{y}_N = \text{WL}\cdot x_2 \\
\]

\[
\text{update}\left(x_2, \hat{y}_1\right) \\
\text{update}\left(x_2, \hat{y}_2\right) \\
\vdots \\
\text{update}\left(x_2, \hat{y}_N\right)
\]
Structure of Online Boosting for Regression

\[ WL^1 \]
\[ \hat{y}_2^1 \]
\[ WL^2 \]
\[ \hat{y}_2^2 \]
\[ \ldots \]
\[ WL^N \]
\[ \hat{y}_2^N \]
Structure of Online Boosting for Regression
Structure of Online Boosting for Regression

\[ WL_1 \]
predict

\[ WL_2 \]
predict

\[ \ldots \]

\[ WL_N \]
predict

\[ x_2 \] \[ \hat{y}_2^1 \] \[ y_2 \]

\[ x_2 \] \[ \hat{y}_2^2 \] \[ \hat{y}_2 \]

\[ x_2 \] \[ \hat{y}_2^N \] \[ \hat{y}_2 \]
Structure of Online Boosting for Regression

\[ WL_1 \] predict

\[ WL_2 \] predict

\[ WL_N \] predict

Booster

\[ x_2 \] \[ \hat{y}_2 \] \[ y_2 \]

\[ (x_2, \hat{y}_2^1) \] \[ WL_1 \] update

\[ (x_2, \hat{y}_2^2) \] \[ WL_2 \] update

\[ (x_2, \hat{y}_2^N) \] \[ WL_N \] update
Structure of Online Boosting

\[ \text{WL}^1 \text{ predict} \]
\[ x_t \xrightarrow{} \hat{y}_t^1 \xrightarrow{} (x_t, \hat{y}_t^1) \xrightarrow{} \text{WL}^1 \text{ update} \]

\[ \text{WL}^2 \text{ predict} \]
\[ x_t \xrightarrow{} \hat{y}_t^2 \xrightarrow{} (x_t, \hat{y}_t^2) \xrightarrow{} \text{WL}^2 \text{ update} \]

\[ \vdots \]

\[ \text{WL}^N \text{ predict} \]
\[ x_t \xrightarrow{} \hat{y}_t^N \xrightarrow{} (x_t, \hat{y}_t^N) \xrightarrow{} \text{WL}^N \text{ update} \]
Constructing the pseudo-labels

Batch boosting:

\[ \tilde{y}_t^1 = y_t \]
Constructing the pseudo-labels

**Batch boosting:**

\[ \tilde{y}_t^1 = y_t \]
\[ \tilde{y}_t^2 = \tilde{y}_t^1 - \eta \tilde{y}_t^1 \]
Constructing the pseudo-labels

Batch boosting:

\[
\tilde{y}_t^1 = y_t \\
\tilde{y}_t^2 = \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 = \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
\vdots
\]

\(\sigma_i^t \in [0, \eta]\) are updated using gradient descent.
Constructing the pseudo-labels

Batch boosting:

\[
\tilde{y}_t^1 = y_t \\
\tilde{y}_t^2 = \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 = \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
\ldots
\]

Online boosting:

\[
\tilde{y}_t^1 = y_t
\]
Constructing the pseudo-labels

Batch boosting:

\[ \tilde{y}_t^1 = y_t \]
\[ \tilde{y}_t^2 = \tilde{y}_t^1 - \eta \hat{y}_t^1 \]
\[ \tilde{y}_t^3 = \tilde{y}_t^2 - \eta \hat{y}_t^2 \]

... 

Online boosting:

\[ \tilde{y}_t^1 = y_t \]
\[ \tilde{y}_t^2 = (1 - \sigma_t^1)\tilde{y}_t^1 + \sigma_t^1 y_t - \eta \hat{y}_t^1 \]
Constructing the pseudo-labels

Batch boosting:

\[
\begin{align*}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
&\quad \vdots
\end{align*}
\]

Online boosting:

\[
\begin{align*}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= (1 - \sigma_t^1)\tilde{y}_t^1 + \sigma_t^1 y_t - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= (1 - \sigma_t^2)\tilde{y}_t^2 + \sigma_t^2 y_t - \eta \hat{y}_t^2 \\
&\quad \vdots
\end{align*}
\]

\(\sigma_t^i \in [0, \eta]\) are updated using gradient descent.
Regret bound

For any \( f \in \text{span}(\mathcal{F}) \),

\[
R'_f(T) \leq \left(1 - \frac{\eta}{\|f\|_1}\right)^N \Delta_0 + O(\|f\|_1 \cdot (\eta T + R(T) + \sqrt{T})),
\]
Regret bound

For any $f \in \text{span}(\mathcal{F})$,

$$R'_f(T) \leq \left( 1 - \frac{\eta}{\|f\|_1} \right)^N \Delta_0 + O(\|f\|_1 \cdot (\eta T + R(T) + \sqrt{T})),$$

where $\Delta_0 := \sum_{t=1}^{T} (y_t - 0)^2 - (y_t - f(x_t))^2$. 
Regret bound

For any $f \in \text{span}(\mathcal{F})$,

$$R'_f(T) \leq \left(1 - \frac{\eta}{\|f\|_1}\right)^N \Delta_0 + O(\|f\|_1 \cdot (\eta T + R(T) + \sqrt{T})),$$

where $\Delta_0 := \sum_{t=1}^{T} (y_t - 0)^2 - (y_t - f(x_t))^2$.

Choosing $\eta \approx \log \frac{N}{N}$, we get $R'_f(T) \to 0$ as $N \to \infty$. 

Regret bound

For any $f \in \text{span}(\mathcal{F})$,

$$R_f'(T) \leq \left(1 - \frac{\eta}{\|f\|_1}\right)^N \Delta_0 + O(\|f\|_1 \cdot (\eta T + R(T) + \sqrt{T})),$$

where $\Delta_0 := \sum_{t=1}^{T} (y_t - 0)^2 - (y_t - f(x_t))^2$.

Choosing $\eta \approx \log \frac{N}{N}$, we get $R_f'(T) \to 0$ as $N \to \infty$.

**Lower bound:** for any online boosting alg, $R_f'(T) \geq \Omega\left(\frac{T}{N}\right)$ for some $f$ in convex hull of $\mathcal{F}$. 
Regret bound

For any $f \in \text{span}(\mathcal{F})$,

$$R'_f(T) \leq \left(1 - \frac{\eta}{\|f\|_1}\right)^N \Delta_0 + O(\|f\|_1 \cdot (\eta T + R(T) + \sqrt{T})),$$

where $\Delta_0 := \sum_{t=1}^{T} (y_t - 0)^2 - (y_t - f(x_t))^2$.

Choosing $\eta \approx \frac{\log N}{N}$, we get $R'_f(T) \to 0$ as $N \to \infty$.

**Lower bound:** for any online boosting alg, $R'_f(T) \geq \Omega\left(\frac{T}{N}\right)$ for some $f$ in convex hull of $\mathcal{F}$.

In **batch** setting, exponentially faster convergence compared to analysis of Zhang and Yu (2005).
Experiments

Setup:

- Implemented within Vowpal Wabbit.
- 14 publicly available data sets
- Parameters $\eta$ and $N$ tuned via progressive validation
- Base learners: VW, Neural Networks, Regression stumps
Experiments

Setup:
- Implemented within Vowpal Wabbit.
- 14 publicly available data sets
- Parameters $\eta$ and $N$ tuned via progressive validation
- Base learners: VW, Neural Networks, Regression stumps

<table>
<thead>
<tr>
<th>Base learner</th>
<th>Average boost</th>
<th>Median boost</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>1.65%</td>
<td>0.03%</td>
</tr>
<tr>
<td>Neural networks</td>
<td>7.88%</td>
<td>0.72%</td>
</tr>
<tr>
<td>Regression stumps</td>
<td>20.22%</td>
<td>10.45%</td>
</tr>
</tbody>
</table>
Conclusions

We propose:

- A natural framework for online boosting.
- An optimal boosting algorithm for classification, Online BBM.
- An adaptive boosting algorithm for classification, AdaBoost.OL.
- An online boosting algorithm for regression.
Conclusions

We propose:
- A natural framework for online boosting.
- An optimal boosting algorithm for classification, Online BBM.
- An adaptive boosting algorithm for classification, AdaBoost.OL.
- An online boosting algorithm for regression.

Future directions:
- **Open problem:** optimal and adaptive boosting algorithm for classification?
- **Open problem:** is our regret bound in the regression setting tight?
- More experimentation and modifications for practical use.