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" $\Rightarrow$ 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" $\Rightarrow$ 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" $\Rightarrow$ 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" $\Rightarrow$ 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" $\Rightarrow$ 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" $\Rightarrow$ 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" $\Rightarrow$ 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[1] Satyen Kale[1] Haipeng Luo[2]

[1]Yahoo! Labs, NYC

[2]Computer Science Department, Princeton University

Batch Boosting

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

Batch Boosting

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

Weak learner (with edge γ):

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

Batch Boosting

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

Weak learner (with edge γ):

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

Strong learner (with error rate ϵ):

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

Batch Boosting

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

Weak learner (with edge γ):

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

$$\Big\Downarrow \text{(Schapire, 1990; Freund, 1995)}$$

Strong learner (with error rate ϵ):

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

Online Boosting

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

Weak Online learner (with edge γ):

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

Strong Online learner (with error rate ϵ):

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

Online Boosting

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

Weak Online learner (with edge γ and excess loss S):

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

Strong Online learner (with error rate ϵ and excess loss S')

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

Online Boosting

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

Weak Online learner (with edge γ and excess loss S):

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

$$\Downarrow \text{ Our result}$$

Strong Online learner (with error rate ϵ and excess loss S')

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

Online Boosting

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

Weak Online learner (with edge γ and excess loss S):

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

$$\Downarrow \text{ Our result}$$

Strong Online learner (with error rate ϵ and excess loss S')

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

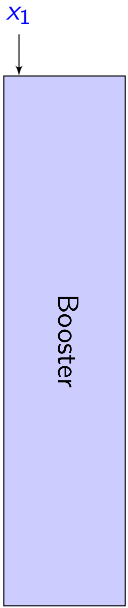
this talk: $S = \frac{1}{\gamma}$ (corresponds to $\sqrt{T}$ regret)

Main Results

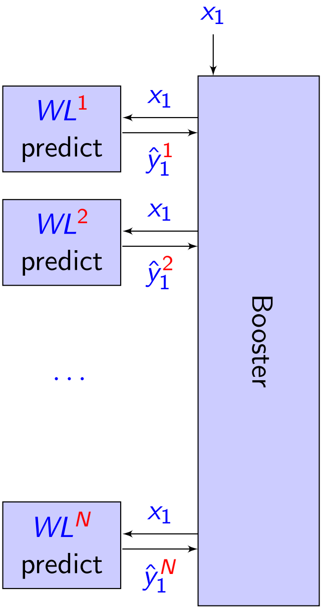
Parameters of interest:
N = number of weak learners (of edge γ) needed to achieve error rate ϵ.
T_ϵ = minimal number of examples s.t. error rate is ϵ.

Algorithm	N	T_ϵ	Optimal?	Adaptive?
Online BBM	$O(\frac{1}{\gamma^2} \ln \frac{1}{\epsilon})$	$\tilde{O}(\frac{1}{\epsilon\gamma^2})$	$\surd$	$\times$
AdaBoost.OL	$O(\frac{1}{\epsilon\gamma^2})$	$\tilde{O}(\frac{1}{\epsilon^2\gamma^4})$	$\times$	$\surd$
Chen et al. (2012)	$O(\frac{1}{\epsilon\gamma^2})$	$\tilde{O}(\frac{1}{\epsilon\gamma^2})$	$\times$	$\times$

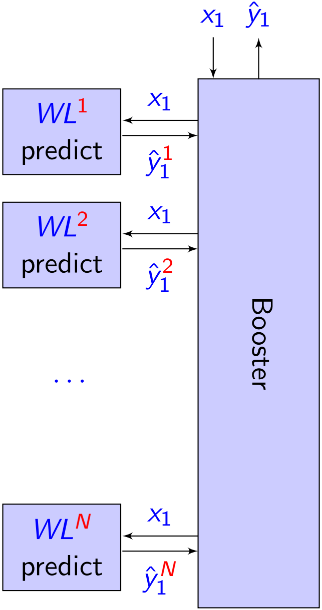
Structure of Online Boosting



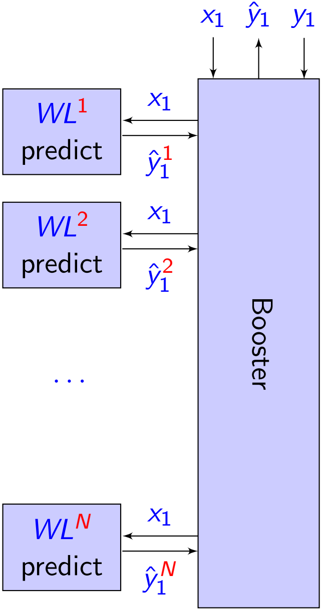
Structure of Online Boosting



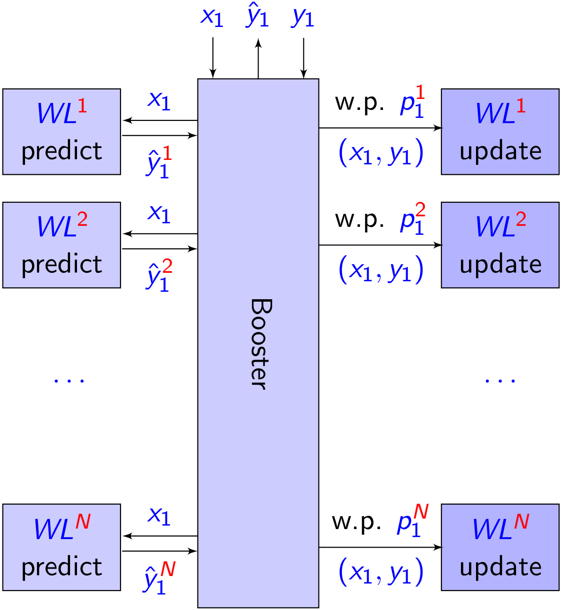
Structure of Online Boosting



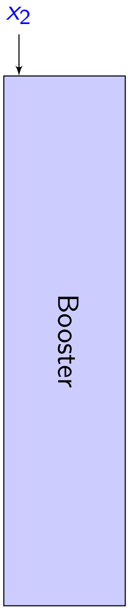
Structure of Online Boosting



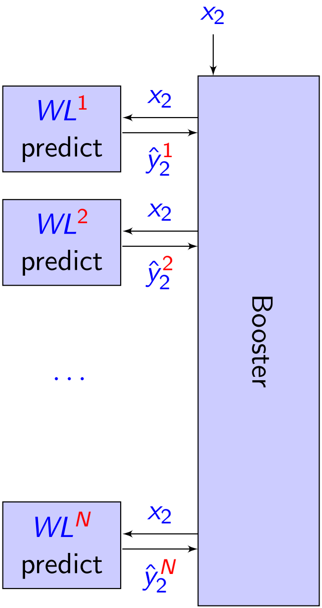
Structure of Online Boosting



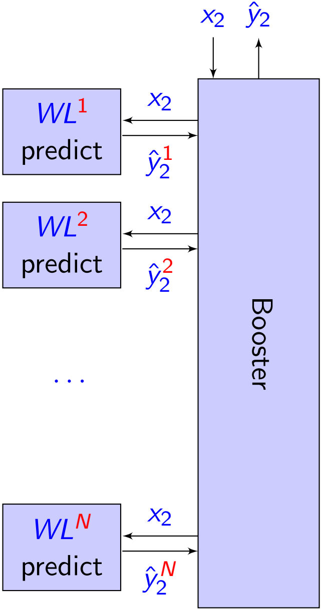
Structure of Online Boosting



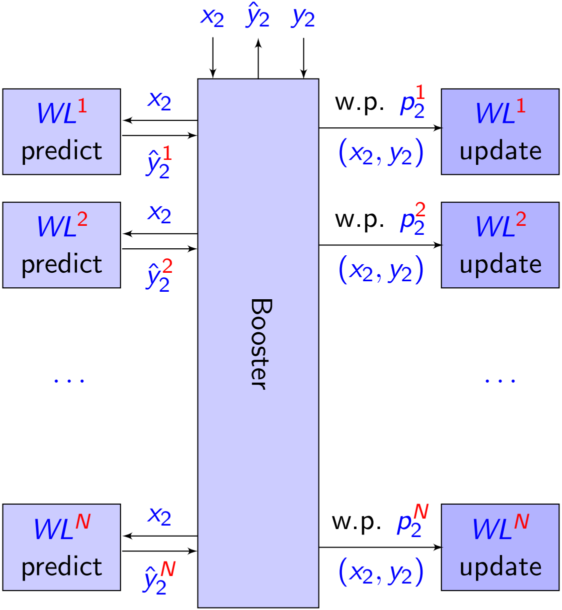
Structure of Online Boosting



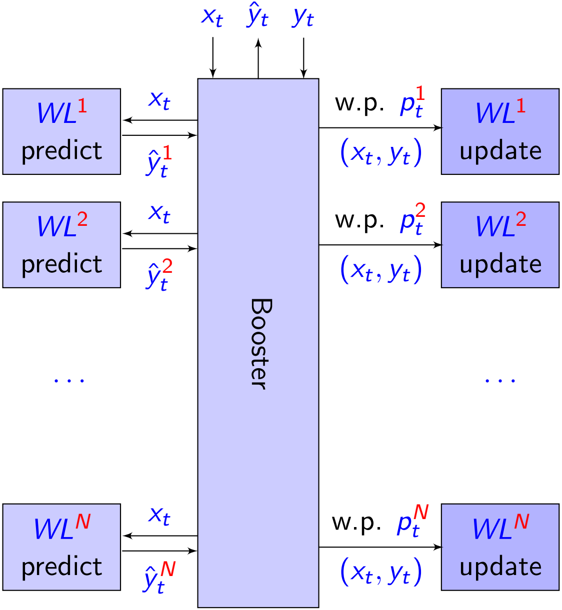
Structure of Online Boosting



Structure of Online Boosting



Structure of Online Boosting



Boosting as a Drifting Game (Schapire, 2001; Luo and Schapire, 2014)

Batch boosting can be analyzed using drifting games.

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.

$$\begin{aligned} \Phi_N(s) &\geq \mathbf{1}\{s \leq 0\}, \\ \Phi_{i-1}(s) &\geq (\tfrac{1}{2} - \tfrac{\gamma}{2})\Phi_i(s-1) + (\tfrac{1}{2} + \tfrac{\gamma}{2})\Phi_i(s+1). \end{aligned}$$

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.

$$\begin{aligned} \Phi_N(s) &\geq \mathbf{1}\{s \leq 0\}, \\ \Phi_{i-1}(s) &\geq (\tfrac{1}{2} - \tfrac{\gamma}{2})\Phi_i(s-1) + (\tfrac{1}{2} + \tfrac{\gamma}{2})\Phi_i(s+1). \end{aligned}$$

Online boosting algorithm using Φ_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.

$$\begin{aligned} \Phi_N(s) &\geq \mathbf{1}\{s \leq 0\}, \\ \Phi_{i-1}(s) &\geq (\tfrac{1}{2} - \tfrac{\gamma}{2})\Phi_i(s-1) + (\tfrac{1}{2} + \tfrac{\gamma}{2})\Phi_i(s+1). \end{aligned}$$

Online boosting algorithm using Φ_i:

- **prediction**: majority vote.
- **update**: $p_t^i = \Pr[(\mathbf{x}_t, y_t) \text{ sent to } WL^i] \propto w_t^i$ where $w_t^i =$ difference in potentials if example is misclassified or not.

Mistake Bound

Generalized drifting games analysis implies

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Online BBM: to get ϵ error rate, needs
$N = O(\frac{1}{\gamma^2} \ln(\frac{1}{\epsilon}))$ weak learners and $T_\epsilon = O(\frac{1}{\epsilon\gamma^2})$ examples. (Optimal)

Drawback of Online BBM

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

- requires γ as a parameter.

Drawback of Online BBM

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

- requires γ 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 γ 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.

Adaptivity via Online Loss Minimization

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 γ_i, then

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

Mistake Bound

If WL^i has edge γ_i, then

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

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

Mistake Bound

If WL^i has edge γ_i, then

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

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

Not optimal but adaptive.

Results

Available in Vowpal Wabbit 8.0.

- command line option: `--boosting`.
- VW as the default "weak" learner (a rather strong one!)

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

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

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ 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.

Regression Setting

Setup:

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

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ 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.

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 $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ 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.

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ is $(y - \hat{y})^2$
- $\mathcal{F}$ is a base class of regressors $f : \mathcal{X} \to [-1, 1]$
- $\mathrm{span}(\mathcal{F})$ 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 η.

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ 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 η.
- Set g to be the constant 0 model.

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ 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 η.
- Set g to be the constant 0 model.
- Repeat for N steps, starting with 0: find

$$f = \arg\min_{f \in \mathcal{F}} \sum_{(\mathbf{x},y) \in S} (\underbrace{y - g(\mathbf{x})}_{\text{residual}} - \eta f(\mathbf{x}))^2$$

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ is $(y - \hat{y})^2$
- $\mathcal{F}$ is a base class of regressors $f : \mathcal{X} \to [-1, 1]$
- $\mathrm{span}(\mathcal{F})$ 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 η.
- Set g to be the constant 0 model.
- Repeat for N steps, starting with 0: find

$$f = \arg\min_{f \in \mathcal{F}} \sum_{(\mathbf{x}, y) \in S} (\underbrace{y - g(\mathbf{x})}_{\text{residual}} - \eta f(\mathbf{x}))^2$$

Regression Setting

Setup:

- Examples $(\mathbf{x}, y) \in \mathcal{X} \times [-1, 1]$
- Loss of predicting $\hat{y}$ for $(\mathbf{x}, y)$ is $(y - \hat{y})^2$
- $\mathcal{F}$ is a base class of regressors $f : \mathcal{X} \to [-1, 1]$
- $\mathrm{span}(\mathcal{F})$ 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 η.
- Set g to be the constant 0 model.
- Repeat for N steps, starting with 0: find

$$f = \arg\min_{f \in \mathcal{F}} \sum_{(\mathbf{x},y)\in S} (\underbrace{y - g(\mathbf{x})}_{\text{residual}} - \eta f(\mathbf{x}))^2$$

and update

$$g \leftarrow g + \eta f.$$

Batch Boosting: Convergence

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

Batch Boosting: Convergence

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

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

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

Batch Boosting: Convergence

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

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

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

Strong learner:

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

Batch Boosting: Convergence

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

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

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

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

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

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

Batch Boosting: Convergence

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

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

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

$$\Downarrow$$

(Friedman, 2001; Mason et al., 2000)
(Zhang and Yu, 2005)

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

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

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

Online Boosting

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

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

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

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

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

$R_f(T) \to 0$ as $N \to \infty$.

Online Boosting

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

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

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

$$\Downarrow \text{ Our result}$$

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

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

$R_f(T) \to 0$ as $N \to \infty$.

Structure of Batch Boosting

Input: batch of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, step-size η

Structure of Batch Boosting

Input: batch of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, step-size η

Set "pseudo-labels" $\tilde{y}_t^1 = y_t$ for all t

Structure of Batch Boosting

Input: batch of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, step-size η

Set "pseudo-labels" $\tilde{y}_t^1 = y_t$ for all t

For $i = 1, 2, \ldots, N$

1. Train weak learner on examples $\{(\mathbf{x}_t, \tilde{y}_t^i) \mid t = 1, 2, \ldots T\}$ with step-size η

Structure of Batch Boosting

Input: batch of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, step-size η

Set "pseudo-labels" $\tilde{y}_t^1 = y_t$ for all t

For $i = 1, 2, \ldots, N$

1. Train weak learner on examples $\{(\mathbf{x}_t, \tilde{y}_t^i) \mid t = 1, 2, \ldots T\}$ with step-size η
2. Obtain predictions $\hat{y}_t^i$ for all t

Structure of Batch Boosting

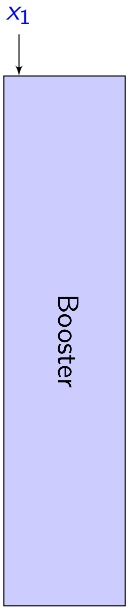
Input: batch of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$, step-size η

Set "pseudo-labels" $\tilde{y}_t^1 = y_t$ for all t

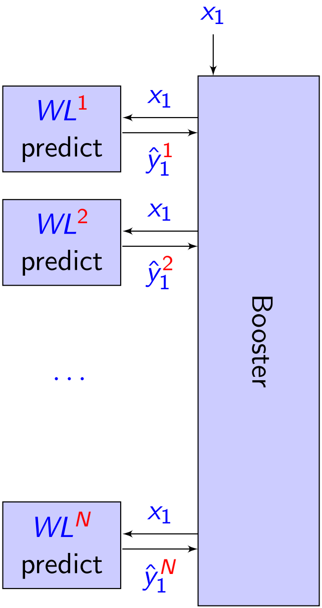
For $i = 1, 2, \ldots, N$

1. Train weak learner on examples $\{(\mathbf{x}_t, \tilde{y}_t^i) \mid t = 1, 2, \ldots T\}$ with step-size η
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

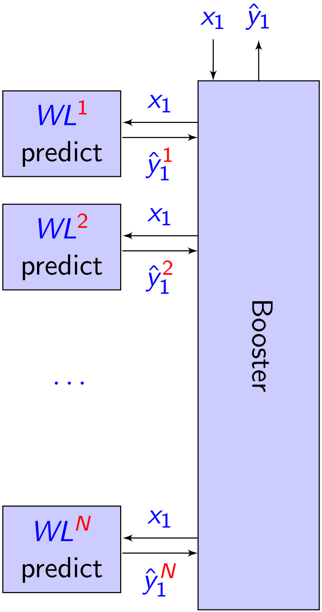
Structure of Online Boosting for Regression



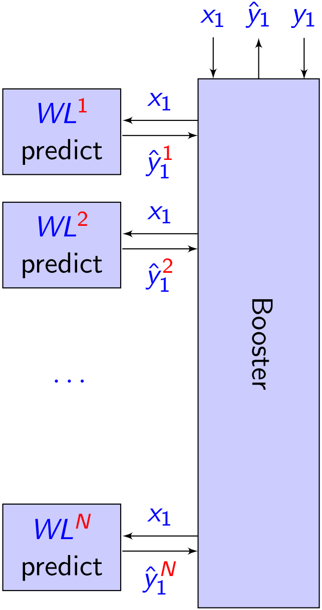
Structure of Online Boosting for Regression



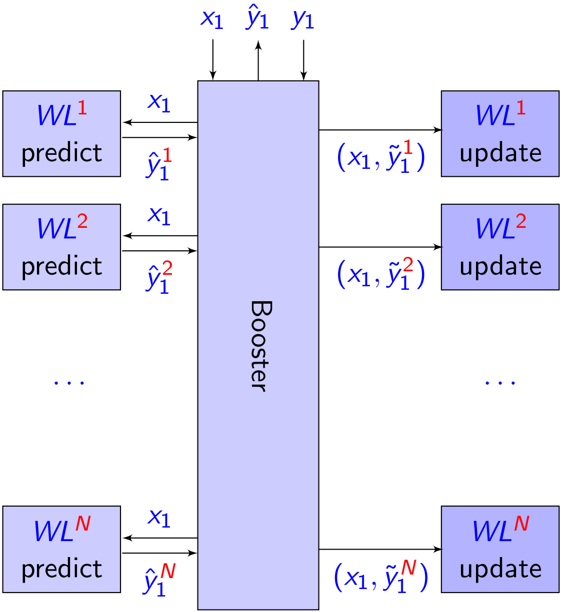
Structure of Online Boosting for Regression



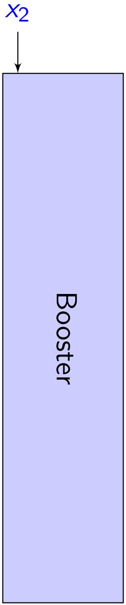
Structure of Online Boosting for Regression



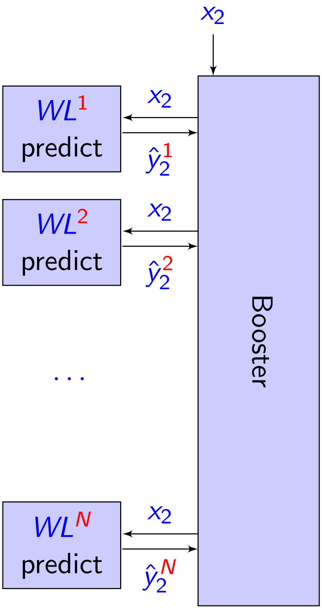
Structure of Online Boosting for Regression



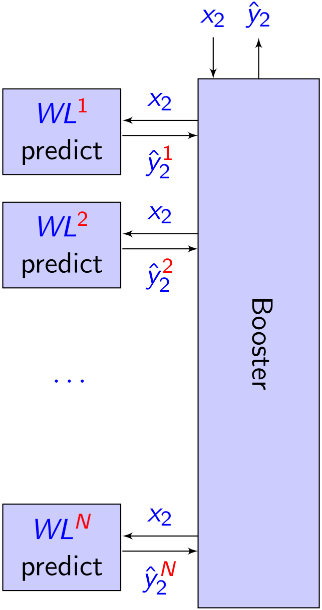
Structure of Online Boosting for Regression



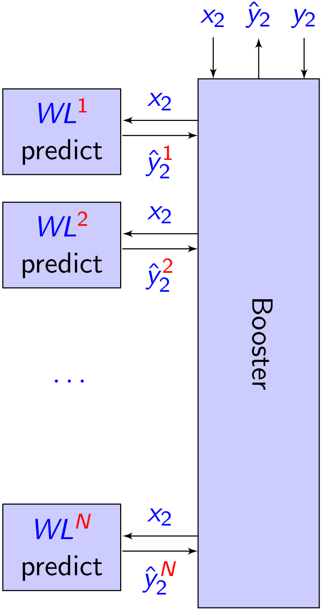
Structure of Online Boosting for Regression



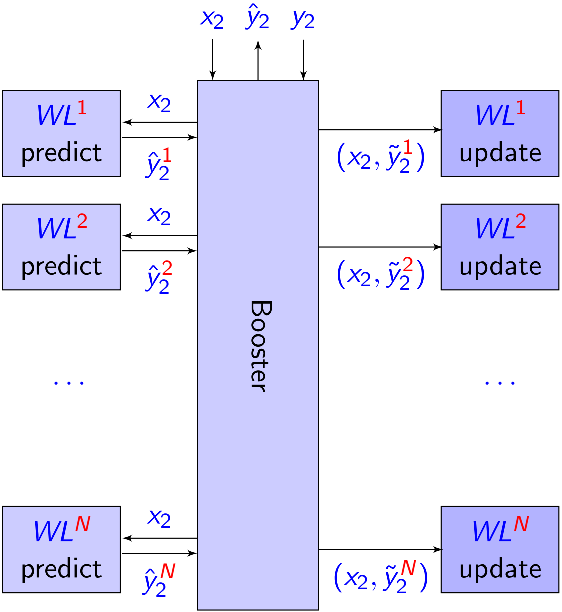
Structure of Online Boosting for Regression



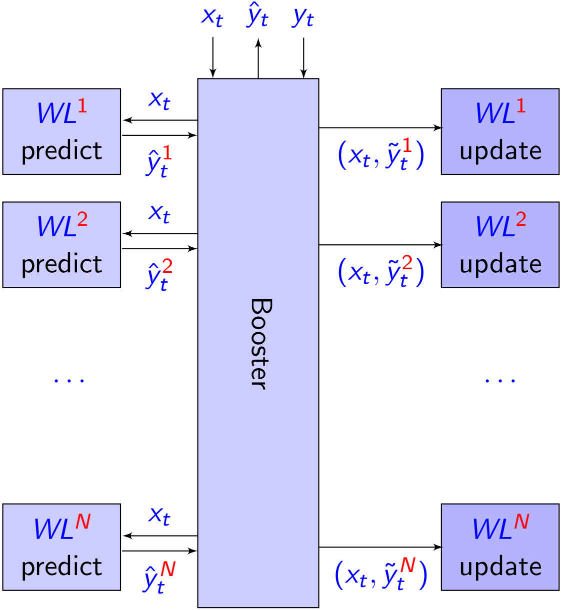
Structure of Online Boosting for Regression



Structure of Online Boosting for Regression



Structure of Online Boosting



Constructing the pseudo-labels

Batch boosting:

$$\tilde{y}_t^1 = y_t$$

Constructing the pseudo-labels

Batch boosting:

$$\begin{aligned}\tilde{y}_t^1 &= y_t \\ \tilde{y}_t^2 &= \tilde{y}_t^1 - \eta \hat{y}_t^1\end{aligned}$$

Constructing the pseudo-labels

Batch boosting:

$$\begin{aligned}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
&\cdots
\end{aligned}$$

Constructing the pseudo-labels

Batch boosting:

$$\begin{aligned}\tilde{y}_t^1 &= y_t \\ \tilde{y}_t^2 &= \tilde{y}_t^1 - \eta\hat{y}_t^1 \\ \tilde{y}_t^3 &= \tilde{y}_t^2 - \eta\hat{y}_t^2 \\ &\ldots\end{aligned}$$

Online boosting:

$$\tilde{y}_t^1 = y_t$$

Constructing the pseudo-labels

Batch boosting:

$$\begin{aligned}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
&\dots
\end{aligned}$$

Online boosting:

$$\begin{aligned}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= (1 - \sigma_t^1)\tilde{y}_t^1 + \sigma_t^1 y_t - \eta \hat{y}_t^1
\end{aligned}$$

Constructing the pseudo-labels

Batch boosting:

$$\begin{aligned}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= \tilde{y}_t^1 - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= \tilde{y}_t^2 - \eta \hat{y}_t^2 \\
&\dots
\end{aligned}$$

Online boosting:

$$\begin{aligned}
\tilde{y}_t^1 &= y_t \\
\tilde{y}_t^2 &= (1 - \sigma_t^1)\tilde{y}_t^1 + \sigma_t^1 y_t - \eta \hat{y}_t^1 \\
\tilde{y}_t^3 &= (1 - \sigma_t^2)\tilde{y}_t^2 + \sigma_t^2 y_t - \eta \hat{y}_t^2 \\
&\dots
\end{aligned}$$

$\sigma_t^i \in [0, \eta]$ are updated using gradient descent.

Regret bound

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

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

Regret bound

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

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

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

Regret bound

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

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

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

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

Regret bound

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

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

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

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, $R'_f(T) \geq \Omega(\frac{T}{N})$ 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 $\Delta_0 := \sum_{t=1}^{T}(y_t - 0)^2 - (y_t - f(\mathbf{x}_t))^2$.

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, $R'_f(T) \geq \Omega(\frac{T}{N})$ 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 η 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 η and N tuned via progressive validation
- Base learners: VW, Neural Networks, Regression stumps

Base learner	**Average boost**	**Median boost**
VW	1.65%	0.03%
Neural networks	7.88%	0.72%
Regression stumps	20.22%	10.45%

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.