Optimal and Adaptive Algorithms for Online Boosting

Alina Beygelzimer[1] **Satyen Kale**[1] Haipeng Luo[2]

[1]Yahoo! Labs, NYC

[2]Computer Science Department, Princeton University

December 11, 2015

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 become infeasible when the amount of data is huge.

Online Boosting: Motivation

Boosting is well studied in the batch setting, but become 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 become 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 become 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.

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 $\mathcal{A}$ predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ 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 $\mathcal{A}$ predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ for example $\mathbf{x}_t$.

Weak learner $\mathcal{A}$ (with edge γ):

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

Batch Boosting

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

Weak learner $\mathcal{A}$ (with edge γ):

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

Strong learner $\mathcal{A}'$ (with any target error rate ϵ):

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon T$$

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 $\mathcal{A}$ predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ for example $\mathbf{x}_t$.

Weak learner $\mathcal{A}$ (with edge γ):

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

$\Downarrow$ Boosting (Schapire, 1990; Freund, 1995)

Strong learner $\mathcal{A}'$ (with any target error rate ϵ):

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon T$$

Online Boosting

Examples $(\mathbf{x}_t, y_t) \in X \times \{-1, 1\}$ arrive online, for $t = 1, \dots, T$.
Learner $\mathcal{A}$ observes $\mathbf{x}_t$ and predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ before seeing y_t.

Weak Online learner $\mathcal{A}$ (with edge γ):

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}(\mathbf{x}_t) \neq y_t\} \leq (\tfrac{1}{2} - \gamma)\, T$$

Strong Online learner $\mathcal{A}'$ (with any target error rate ϵ):

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon\, T$$

Online Boosting

Examples $(\mathbf{x}_t, y_t) \in X \times \{-1, 1\}$ arrive online, for $t = 1, \ldots, T$.
Learner $\mathcal{A}$ observes $\mathbf{x}_t$ and predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ before seeing y_t.

Weak Online learner $\mathcal{A}$ (with edge γ and excess loss S):

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

Strong Online learner $\mathcal{A}'$ (with any target error rate ϵ and excess loss S')

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon T + S'$$

Online Boosting

Examples $(\mathbf{x}_t, y_t) \in X \times \{-1, 1\}$ arrive online, for $t = 1, \ldots, T$.
Learner $\mathcal{A}$ observes $\mathbf{x}_t$ and predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ before seeing y_t.

Weak Online learner $\mathcal{A}$ (with edge γ and excess loss S):

$$\textstyle\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}(\mathbf{x}_t) \neq y_t\} \leq (\frac{1}{2} - \gamma) T + S$$

$\Downarrow$ Online Boosting (our result)

Strong Online learner $\mathcal{A}'$ (with any target error rate ϵ and excess loss S')

$$\textstyle\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon T + S'$$

Online Boosting

Examples $(\mathbf{x}_t, y_t) \in X \times \{-1, 1\}$ arrive online, for $t = 1, \ldots, T$.
Learner $\mathcal{A}$ observes $\mathbf{x}_t$ and predicts $\mathcal{A}(\mathbf{x}_t) \in \{-1, 1\}$ before seeing y_t.

Weak Online learner $\mathcal{A}$ (with edge γ and excess loss S):

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

$\Downarrow$ Online Boosting (our result)

Strong Online learner $\mathcal{A}'$ (with any target error rate ϵ and excess loss S')

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \epsilon T + S'$$

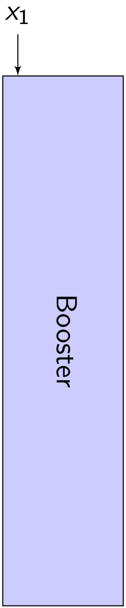
this talk: $S = \frac{1}{\gamma}$ (corresponds to $\sqrt{T}$ regret)

Main Results

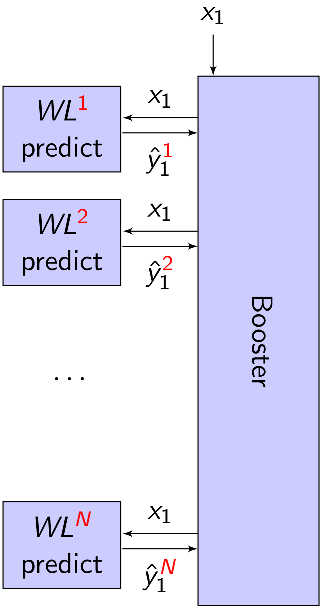
Parameters of interest:
N = number of weak learners (of edge γ) needed to achieve error rate ϵ.
T_ϵ = minimal number of examples s.t. error rate is ϵ.

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)	$O(\frac{1}{\epsilon\gamma^2})$	$\tilde{O}(\frac{1}{\epsilon\gamma^2})$	$\times$	$\times$

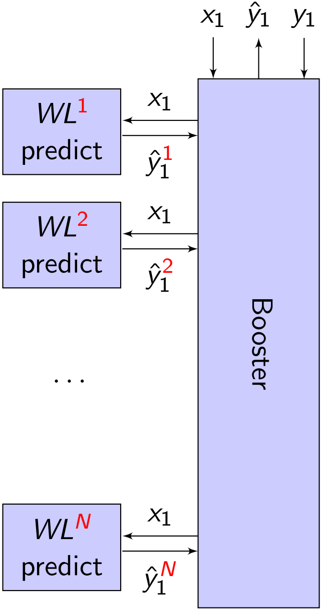
Structure of Online Boosting



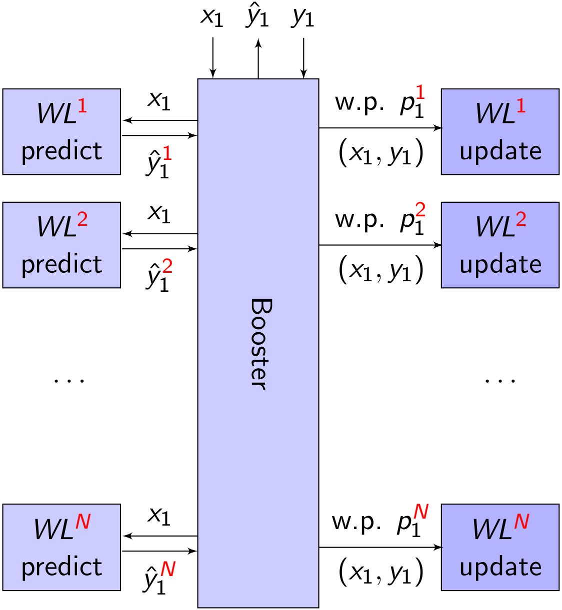
Structure of Online Boosting



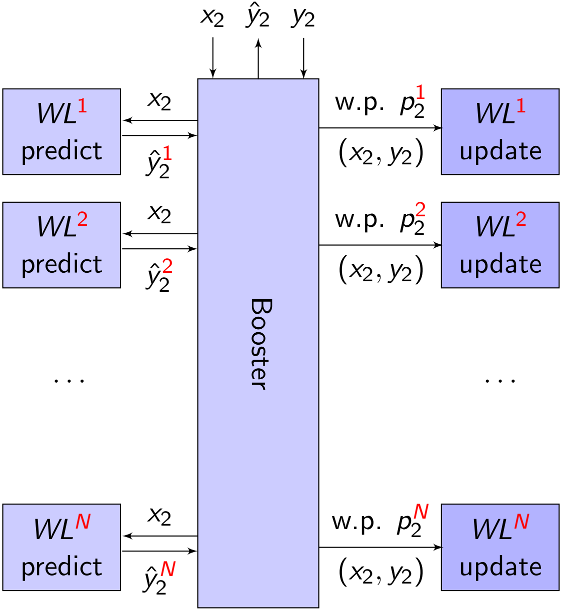
Structure of Online Boosting



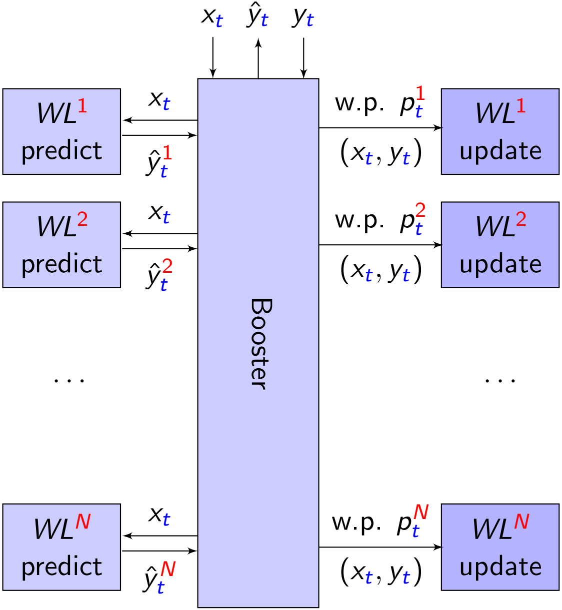
Structure of Online Boosting



Structure of Online Boosting



Structure of Online Boosting



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

Batch boosting can be analyzed using drifting game.

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

Batch boosting can be analyzed using drifting game.

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

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

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

Batch boosting can be analyzed using drifting game.

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

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

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 game.

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

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

Online boosting algorithm using Φ_i:

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

Mistake Bound

Generalized drifting games analysis implies

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

Mistake Bound

Generalized drifting games analysis implies

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

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

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

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

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

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$] \leq \frac{4}{\sqrt{N-i}}$

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.
- Adapts to *easy data*

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 this 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 this 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.

Intuition and main ideas

- Classifier f with real-valued output $f(x)$: predict $\operatorname{sign}(f(x))$
- Logistic loss $\ln(1 + \exp(-f(x)y))$: surrogate for $1\{\operatorname{sign}(f(x)) \neq y\}$

Intuition and main ideas

- Classifier f with real-valued output $f(x)$: predict $\mathrm{sign}(f(x))$
- Logistic loss $\ln(1+\exp(-f(x)y))$: surrogate for $1\{\mathrm{sign}(f(x)) \neq y\}$
- In batch setting (AdaBoost.L):
  - for each i, add output of weak learner with step-size α found by line search to minimize logistic loss

Intuition and main ideas

- Classifier f with real-valued output $f(x)$: predict $\text{sign}(f(x))$
- Logistic loss $\ln(1 + \exp(-f(x)y))$: surrogate for $1\{\text{sign}(f(x)) \neq y\}$
- In batch setting (AdaBoost.L):
  - for each i, add output of weak learner with step-size α found by line search to minimize logistic loss
  - final prediction is weighted majority with weights α

Intuition and main ideas

- Classifier f with real-valued output $f(x)$: predict $\text{sign}(f(x))$
- Logistic loss $\ln(1 + \exp(-f(x)y))$: surrogate for $1\{\text{sign}(f(x)) \neq y\}$
- In batch setting (AdaBoost.L):
  - for each i, add output of weak learner with step-size α found by line search to minimize logistic loss
  - final prediction is weighted majority with weights α
- In online setting (AdaBoost.OL):
- for each i, search for step-size α using *online gradient descent* over sequence of T data points

Intuition and main ideas

- Classifier f with real-valued output $f(x)$: predict $\text{sign}(f(x))$
- Logistic loss $\ln(1 + \exp(-f(x)y))$: surrogate for $1\{\text{sign}(f(x)) \neq y\}$
- In batch setting (AdaBoost.L):
  - for each i, add output of weak learner with step-size α found by line search to minimize logistic loss
  - final prediction is weighted majority with weights α
- In online setting (AdaBoost.OL):
- for each i, search for step-size α using *online gradient descent* over sequence of T data points
- for each data point, final prediction is weighted majority with weights given by current α's

Mistake Bound

If WL^i has edge γ_i, then

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)$$

Mistake Bound

If WL^i has edge γ_i, then

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)$$

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

$$\sum_{t=1}^{T} \mathbf{1}\{\mathcal{A}'(\mathbf{x}_t) \neq y_t\} \leq \frac{2T}{\sum_i \gamma_i^2} + \tilde{O}\left(\frac{N^2}{\sum_i \gamma_i^2}\right)$$

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

Conclusions

We propose

- A natural framework of online boosting.
- An optimal algorithm Online BBM.
- An adaptive algorithm AdaBoost.OL.

Conclusions

We propose

- A natural framework of online boosting.
- An optimal algorithm Online BBM.
- An adaptive algorithm AdaBoost.OL.

Open problem: optimal and adaptive algorithm?