The Weighted Majority Algorithm

Fabrice Zapfack

3rd October, 2016

Plan

1. Introduction
2. The Weighted Majority Algorithm
3. Improvements
4. Applications

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

Halving Algorithm

- At each step predict the majority vote
- Keep only the functions that are consistent

Theorem

Let m be the number of mistakes when we apply Halving on $\mathcal{S}$ with the pool $\mathcal{A}$.

$$m \leq log_2(|A|)$$

ATTENTION - Works only in the realisable case.

Plan

1 Introduction

2 The Weighted Majority Algorithm

3 Improvements

4 Applications

Framework for WMA

WMA - Given a pool of algorithms $\mathcal{A}$ where one of them performs well, we will create a compound algorithm based on a weighted voting of the algorithms of $\mathcal{A}$.

Different frameworks

- First case - Binary predictions, finite cardinal of $\mathcal{A}$
- Second case -
- Third case

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

Bound on the number of mistakes

Notations

- $\mathcal{S}$: sequence of instances and binary labels
- w_{init} : initial total weight of the algorithms in the pool
- w_{fin} : final total weight of the algorithms in the pool
- q_0 : total weight of the algorithms that predict 0
- q_1 : total weight of the algorithms that predict 1
- β : weight multiplication factor in case of a mistake $(0 \leq \beta < 1)$

Theorem

Let m be the number of mistakes when we apply WMA on $\mathcal{S}$ with the pool $\mathcal{A}$.

$$m \leq \frac{\log(w_{init}/w_{fin})}{\log(2/(1+\beta))}$$

Proof sketch

- Step 1 - Prove that if in a trial WM makes a mistake, the sum of weights before the trial is greater than $u = \frac{1+\beta}{2}$ times the sum of weights after
- Step 2 - Recursively, we get that $w_{init} u^m \geq w_{fin}$
- Step 3 - Take the log of the above inequality to conclude

Proof of step 1 - Let us suppose wlog that the learner predicted 0 which was the wrong binary label in this trial.

- Total weight before : $q_0 + q_1$. Since 0 was predicted, $q_0 \geq q_1$
- Total weight after :
$\beta q_0 + q_1 \leq \beta q_0 + q_1 + \frac{1-\beta}{2}(q_0 + q_1) \leq \frac{1+\beta}{2}(q_0 + q_1)$

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

What motivates this modified version?

- First version : selects a group of "good" algorithms and makes other weights shrink
- Not very good for adaptation : if throughout the trials another group becomes better it will not be seen because of its small weight
- Idea : Do not allow an algorithm's weight to be lower that $\frac{\gamma}{|\mathcal{A}|}$ times the total weight of $\mathcal{A}$

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

Assumptions for WMG and WMC

- Predictions of algorithms in the pool in [0,1] for both
- WMG predictions are binary
- WMC predictions are in [0,1]
- Labels are binary for WMG
- Labels are in [0,1] for WMC

Update step

- Updates at every trial for WMC
- Updates either at every trial or only when a mistake occurs

Notations

- *Update-trial j* : j-th trial in which an update occurs, $j = 1..t$, $|\mathcal{A}| = n$
- $x_i^{(j)}$: prediction of the i-th algorithm in update-trial j
- $\lambda^{(j)}$: prediction of the master (compound) algorithm in update-trial j
- $\rho^{(j)}$: label of update-trial j
- $w_1^{(j)}, ..., w_n^{(j)}$: weights at the beginning of update-trial j
- $s^{(j)} = \sum_{i=1}^{n} w_i^{(j)}, \gamma^{(j)} = \frac{\sum_{i=1}^{n} w_i^{(j)} x_i^{(j)}}{s^{(j)}}$

Predictions

- WMC : $\lambda^{(j)} = \gamma^{(j)}$
- WMG : $\lambda^{(j)} = 1$ if $\gamma^{(j)} \geq \frac{1}{2}$ and $\lambda^{(j)} = 0$ if $\gamma^{(j)} < \frac{1}{2}$

Bound for WMG

Definition (Update step for WMG and WMC)

$w_i^{(j+1)} = F w_i^{(j)}$ *where* $F = F(\beta, x_i^{(j)}, \rho^{(j)})$ *satisfies :*

$$\beta^{|x_i^{(j)} - \rho^{(j)}|} \leq F \leq 1 - (1-\beta)|x_i^{(j)} - \rho^{(j)}|$$

Theorem (Bound for WMG)

Let m *be the number of mistakes when running WMG on* $\mathcal{S}$ *with* $0 \leq \beta < 1$.

$$m \leq \frac{\log(w_{init}/w_{fin})}{\log(2/(1+\beta))}$$

Bound for WMG - Proof

Let us start by proving that we can always find an update factor F :

Lemma

For $\beta \geq 0$ and $0 \leq r \leq 1$: $\beta^r \leq 1 + r(\beta - 1)$

PROOF - Convexity inequality on $r \mapsto \beta^r$
In the following lemma, we will assume that $w_i^{(1)} > 0$, $\rho^{(j)} \leq 1$ and $0 \leq x_i^{(j)} \leq 1$ for $i = 1..n, j = 1..t$.

Lemma (Crux lemma)

Let us also assume that for $i = 1..n, j = 1..t, w_i^{(j+1)} \leq w_i^{(j)}(1 - (1-\beta)|x_i^{(j)} - \rho^{(j)}|)$. *If* $\beta = 0$ *and* $|\gamma^{(j)} - \rho^{(j)}| = 1$ *for some* j, *then* $w_{fin} = 0$. *Otherwise,*

$$\log(\frac{w_{fin}}{w_{init}}) \leq \sum_{j=1}^{t} \log(1 - (1-\beta)|\gamma^{(j)} - \rho^{(j)}|)$$

PROOF.

First case - If $\beta = 0$ and $|\gamma^{(j)} - \rho^{(j)}| = 1$ for some j. Then

$$|\frac{\sum_i w_i^{(j)} x_i^{(j)}}{\sum_i w_i^{(j)}} - \frac{\sum_i w_i^{(j)} \rho^{(j)}}{\sum_i w_i^{(j)}}| = 1$$

$$\Rightarrow |\frac{\sum_i w_i^{(j)}(x_i^{(j)} - \rho^{(j)})}{\sum_i w_i^{(j)}}| = 1$$

Proof of the Crux lemma (..page 2..)

$$\Rightarrow \frac{\sum_i w_i^{(j)} |x_i^{(j)} - \rho^{(j)}|}{\sum_i w_i^{(j)}} \geq 1$$

Since $x_i^{(j)}, \rho^{(j)} \in [0,1]$, $|x_i^{(j)} - \rho^{(j)}| \leq 1$, so $\frac{\sum_i w_i^{(j)} |x_i^{(j)} - \rho^{(j)}|}{\sum_i w_i^{(j)}}$ can only be greater than 1 if $|x_i^{(j)} - \rho^{(j)}| = 1$ for $i = 1..n$, so we have to use the update factor $(\beta = 0)$ and $w_i^{(j+1)} = 0$ for $i = 1..n$ so $w_{fin} = 0$.

Second case - Using the convexity inequality from the proof of a previous lemma :

$$s^{(j+1)} \leq \sum_{i=1}^{n} w_i^{(j)} (1 - (1-\beta)|x_i^{(j)} - \rho^{(j)}|) = s^{(j)} - (1-\beta) \sum_{i=1}^{n} w_i^{(j)} |x_i^{(j)} - \rho^{(j)}|$$

Proof of the Crux lemma (..end)

Using the triangular inequality:

$$\begin{aligned} s^{(j)} - (1-\beta)\sum_{i=1}^{n} w_i^{(j)}|x_i^{(j)} - \rho^{(j)}| &\leq s^{(j)} - (1-\beta)|\sum_{i=1}^{n} w_i^{(j)}(x_i^{(j)} - \rho^{(j)})| \\ &= s^{(j)} - (1-\beta)|\sum_{i=1}^{n} \gamma^{(j)} s^{(j)} - \rho^{(j)} s^{(j)}| \\ &= s^{(j)}(1-(1-\beta)|\gamma^{(j)} - \rho^{(j)}|) \end{aligned}$$

Recursively, we get :

$$s^{(t+1)} \leq s^{(1)} \prod_{j=1}^{t} (1-(1-\beta)|\gamma^{(j)} - \rho^{(j)}|)$$

Proof of the WMG bound

Proof of the WMG bound. First case - $\beta = 0 \Rightarrow w_{fin} = 0$ so the bound becomes ... Second case - Let $m^{(j)} = 1$ if WMG makes a mistake in update-trial j, 0 otherwise. $m = \sum_{j=1}^{t} m^{(j)}$.
Since $\log(1-(1-\beta)|\gamma^{(j)} - \rho^{(j)}|) \leq 0$,

$$\sum_{j=1}^{t} \log(1-(1-\beta)|\gamma^{(j)}-\rho^{(j)}|) \leq \sum_{j \text{ s.t. } m^{(j)}=1} \log(1-(1-\beta)|\gamma^{(j)}-\rho^{(j)}|)$$

- If $\gamma^{(j)} < \frac{1}{2}$, $m^{(j)} = 1 \Rightarrow \rho^{(j)} = 1 \Rightarrow |\gamma^{(j)} - \rho^{(j)}| \geq \frac{1}{2}$
- If $\gamma^{(j)} \geq \frac{1}{2}$, $m^{(j)} = 1 \Rightarrow \rho^{(j)} = 0 \Rightarrow |\gamma^{(j)} - \rho^{(j)}| \geq \frac{1}{2}$

Proof of the WMG bound (..end)

So,

$$\sum_{j \text{ s.t. } m^{(j)}=1} \log(1-(1-\beta)|\gamma^{(j)}-\rho^{(j)}|) \leq m\log(1-\frac{1}{2}(1-\beta)) = m\log(\frac{1}{2}+\frac{1}{2}\beta)$$

We can use the Crux lemma and get :

$$\log(\frac{W_{fin}}{W_{init}}) \leq m\log(\frac{1+\beta}{2})$$

Bound for WMC

Definition (Loss m for WMC)

For continuous predictions, the loss is defined by :

$$m = \sum_{j=1}^{t} |\lambda^{(j)} - \rho^{(j)}|$$

Theorem

Let $\mathcal{S}$ be any sequence of instances and labels, with labels in [0,1]. Let m be the total loss for the WMC.

$$m \leq \frac{\log(w_{init}/w_{fin})}{1-\beta}$$

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

Assumptions

- Predictions of pool members are in [0,1]
- Prediction of WMR is binary but probabilistic
- Labels associated with instances are binary

Prediction of WMR

- WMR predicts 1 with probability $\gamma^{(j)}$
- If pool members' predictions are binary, $\gamma^{(j)} = \frac{q_1}{q_0+q_1}$.

Update criterion

- Update in every trial

Update step

- Same update step as WMG and WMC.

Plan

1. Introduction
2. The Weighted Majority Algorithm
3. Improvements
4. Applications

Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

EWA

- fix $p_1 = \pi$ an arbitrary probability distribution on $\mathbb{R}^M$
- $\hat{y}_t = \int f_\theta(x_t) p_t(\mathrm{d})\theta$ and once y_t is revealed,

$$p_{t+1}(\mathrm{d}\theta) = \frac{\exp\left(-\eta\ell(y_t, f_\theta(x_t))\right) p_t(\mathrm{d}\theta)}{\int_{\mathbb{R}^M} \exp\left(-\eta\ell(y_t, f_\alpha(x_t))\right) p_t(\mathrm{d}\alpha)}.$$

Theorem

Taking $\eta = 2\sqrt{\frac{2\log(M)}{TC^2}}$ leads to a regret in

$$\mathcal{R}_T(\{f_1, \ldots, f_M\}) \leq C\sqrt{\frac{T\log(M)}{2}}.$$

ATTENTION - Applicable to L-type, C-type, MS-type aggregation

Plan

1 Introduction

2 The Weighted Majority Algorithm

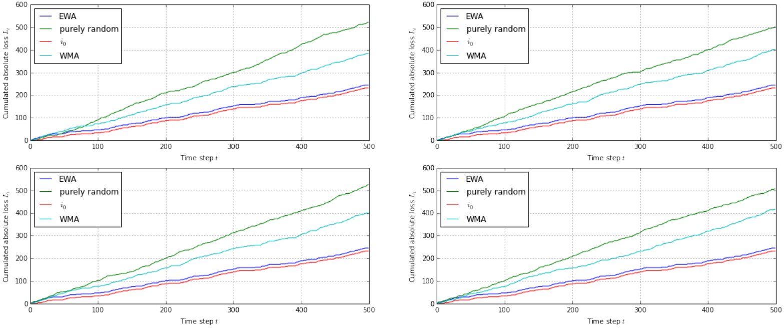
3 Improvements

4 Applications

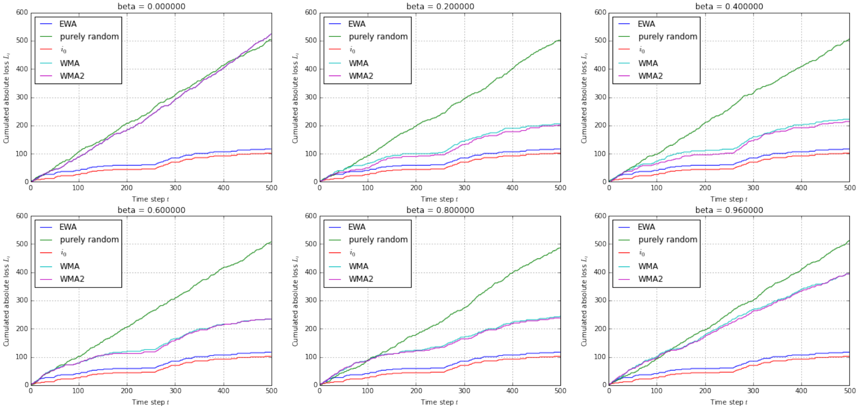
Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

Application



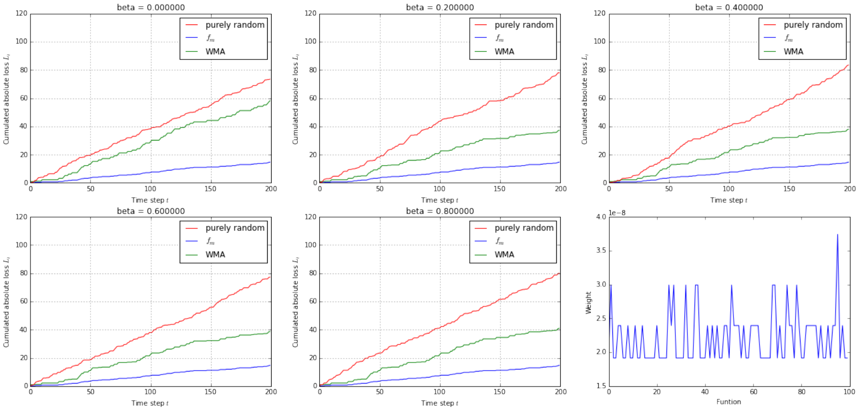
Different initial weights



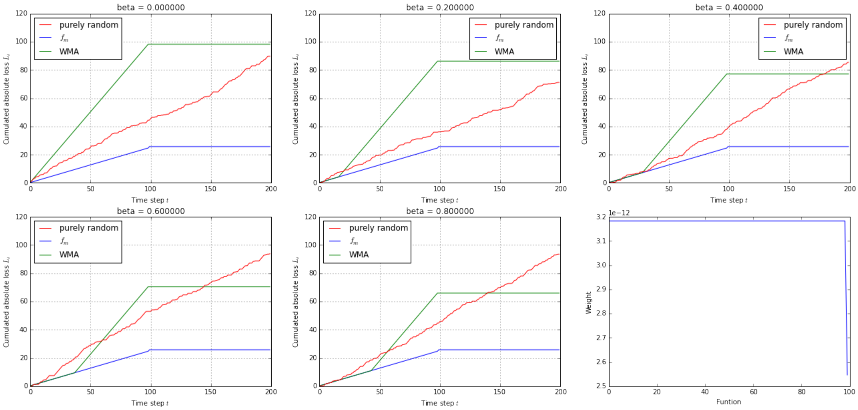
Plan

1. Introduction
   - Halving Algorithm
2. The Weighted Majority Algorithm
   - Classic version - Binary predictions, finite cardinal
   - Modified version
   - Generalized version - WMG and WMC
   - Randomized version
3. Improvements
   - Exponentially Weighted Aggregation (EWA)
4. Applications
   - Applicable case
   - Non-applicable case

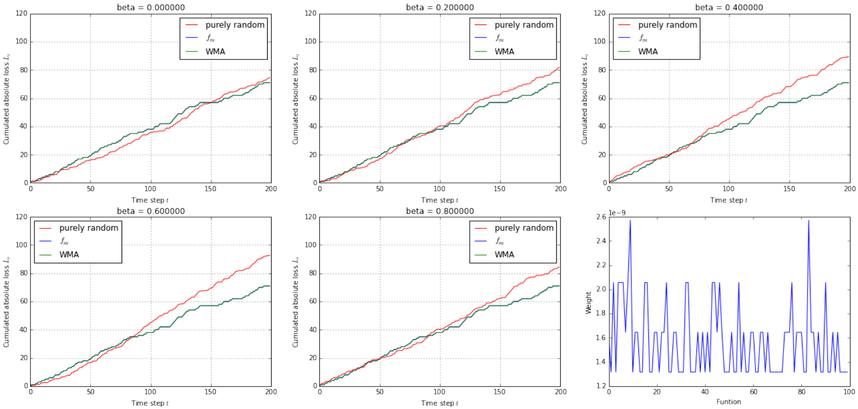
Application



Different initial weights



Application



Different initial weights