Lecture 26: Multiplicative Weights (Continued)

Michael Dinitz

December 4, 2025
601.433/633 Introduction to Algorithms
Slides by Michael Dinitz and Jessica Sorrell

Introduction

Last time, we introduced the online learning setting and the multiplicative weights algorithm applied to online learning.

Introduction

Last time, we introduced the online learning setting and the multiplicative weights algorithm applied to online learning.

Today, we'll introduce the supervised learning setting and discuss applications of the multiplicative weights algorithm to this setting.

Supervised Learning Intro

Trying to learn "Yes/No" labels

- Given a photo, does it have a dog in it?
- Given an email, is it spam?

Given some labeled data. Create a good prediction rule (*hypothesis*) for future data.

Supervised Learning Intro

Trying to learn "Yes/No" labels

- Given a photo, does it have a dog in it?
- Given an email, is it spam?

Given some labeled data. Create a good prediction rule (*hypothesis*) for future data.

Example: spam

- Want to create a rule (hypothesis) that will tell us whether an email is spam
- Given some example emails with labels (Yes / No, Spam / Not Spam)

Example

sales	apply	Mr.	bad spelling	known-sender	spam?
Y	N	Y	Y	N	Y
N	N	N	Y	Y	N
N	Y	N	N	N	Y
Y	N	N	N	Y	N
N	N	Y	N	Y	N
Y	N	N	Y	N	Y
N	N	Y	N	N	N
N	Y	N	Y	N	Y

Example

sales	apply	Mr.	bad spelling	known-sender	spam?
Y	N	Y	Y	N	Y
N	N	N	Y	Y	N
N	Y	N	N	N	Y
Y	N	N	N	Y	N
N	N	Y	N	Y	N
Y	N	N	Y	N	Y
N	N	Y	N	N	N
N	Y	N	Y	N	Y

Reasonable hypothesis:
spam if not known-sender
AND (apply OR sales)

Questions

Question 1: Can we efficiently find working hypothesis for given labeled data?

- Mainly about efficiency; like many of the problems we've talked about
- Depends on what kinds of hypotheses we're looking for (structure and quality)

Question 2: Can we be confident that our hypothesis will do well in the future?

- Not primarily about efficiency; about quality
- Requires knowing something about the future!
- Core of machine learning: use the past to make predictions about the future

Questions

Question 1: Can we efficiently find working hypothesis for given labeled data?

- Mainly about efficiency; like many of the problems we've talked about
- Depends on what kinds of hypotheses we're looking for (structure and quality)

Question 2: Can we be confident that our hypothesis will do well in the future?

- Not primarily about efficiency; about quality
- Requires knowing something about the future!
- Core of machine learning: use the past to make predictions about the future

Will mostly focus on Question 1 today and ignore Question 2 (but consider Prof. Arora's course on machine learning theory if you're curious!)

Formalization: Beginning

Given sample set $S = \{(x^1, y^1), \ldots (x^m, y^m)\}$. Size m called the *sample complexity*

- Each x^i from domain X
- $y^i = f(x^i)$ for some unknown f ($f : X \to \{0, 1\}$ from a class of Boolean functions $\mathcal{F}$)

Our goal today: compute hypothesis h with low *error* on S:

$$err(h) := \frac{1}{|S|} \sum_{i \in [m]} \mathbb{1}[h(x^i) \neq f(x^i)] \leq \epsilon$$

indicator

Formalization: Beginning

Given sample set $S = \{(x^1, y^1), \ldots (x^m, y^m)\}$. Size m called the *sample complexity*

- Each x^i from domain X
- $y^i = f(x^i)$ for some unknown f ($f : X \to \{0, 1\}$ from a class of Boolean functions $\mathcal{F}$)

Our goal today: compute hypothesis h with low *error* on S:

$$err(h) := \frac{1}{|S|} \sum_{i \in [m]} \mathbb{1}[h(x^i) \neq f(x^i)] \leq \epsilon$$

Could just return a lookup table for S and return random values on unseen data, but this would definitely not work well in the future. We want to consider approaches that have some hope of generalizing to unseen data (though we won't prove they do today).

Reducing strong learning to weak learning

Strong Learner

An algorithm $\mathcal{L}$ is a **strong learner** for a function class $\mathcal{F}$ if for any dataset S labeled by a function $f \in \mathcal{F}$, it can find a hypothesis h with error at most ϵ.

$y_i = f(x_i)$

$$\frac{1}{m} \sum_{i \in [m]} \mathbb{1}[h(x^i) \neq y^i] \leq \epsilon$$

Reducing strong learning to weak learning

Strong Learner

An algorithm $\mathcal{L}$ is a **strong learner** for a function class $\mathcal{F}$ if for any dataset S labeled by a function $f \in \mathcal{F}$, it can find a hypothesis h with error at most ϵ.

$$\frac{1}{m} \sum_{i \in [m]} \mathbb{1}[h(x^i) \neq y^i] \leq \epsilon$$

Weak Learner

Let $\gamma \in (0, \frac{1}{2})$. An algorithm WkL is a γ-**weak learner** for $\mathcal{F}$ if for any distribution D over a dataset S labeled by a function $f \in \mathcal{F}$, it can find a hypothesis h that is slightly better than random guessing on that distribution.

$$\Pr_{(x^i, y^i) \sim D}[h(x^i) \neq y^i] \leq \frac{1}{2} - \gamma$$

The MW Boosting Algorithm

We can construct a strong learner from a weak learner by iteratively reweighting the sample using multiplicative weights.

Algorithm MW Boosting

Let $S = \{(x^1, y^1), \ldots (x^m, y^m)\}$ be a dataset.
Initialize weights $w_i^1 = 1$ for all $i \in [m]$.
for $t = 1, \ldots, T$ **do**
    Define distribution $D^t(i) = \frac{w_i^t}{\sum_j w_j^t}$.
    Run *WkL* on (S, D^t) to get hypothesis h_t.
    Define loss $\ell_j^t = 1$ if $h_t(x_j) = y_j$, and 0 otherwise.
    Update weights: $w_j^{t+1} \leftarrow w_j^t \cdot e^{-\gamma \ell_j^t}$.
end for
return $h(x)$ = majority vote of $\{h_1(x), \ldots, h_T(x)\}$

Boosting Intuition

Compare with multiplicative weights for online learning

- We use the same exponential update rule
- But reweight *examples* rather than *actions*

Boosting Intuition

Compare with multiplicative weights for online learning

- We use the same exponential update rule
- But reweight *examples* rather than *actions*
- Examples that are **correctly** classified get a **high loss** ($\ell_j^t = 1$) and are **down-weighted**

Boosting Intuition

Compare with multiplicative weights for online learning

- We use the same exponential update rule
- But reweight *examples* rather than *actions*
- Examples that are **correctly** classified get a **high loss** ($\ell_j^t = 1$) and are **down-weighted**
- Examples that are **incorrectly** classified get a **low loss** ($\ell_j^t = 0$) and their weight stays the same

Boosting Intuition

Compare with multiplicative weights for online learning

- We use the same exponential update rule
- But reweight *examples* rather than *actions*
- Examples that are **correctly** classified get a **high loss** ($\ell_j^t = 1$) and are **down-weighted**
- Examples that are **incorrectly** classified get a **low loss** ($\ell_j^t = 0$) and their weight stays the same

We want to force the weak learner to focus on the "hard" examples that we have misclassified so far. Since the weak learner is guaranteed to do better than random guessing on the new weighted distribution, we make progress on these hard examples in expectation, so long as they have enough weight.

Main Theorem

Theorem

*Let $\varepsilon \in (0,1)$ and $\gamma \in (0, \frac{1}{2})$. Let $\mathcal{F}$ be a class of Boolean functions and let S be a dataset labeled by some $f \in \mathcal{F}$. Given access to a γ-weak learner **WkL** and run for $T \geq \frac{2\log(1/\varepsilon)}{\gamma^2}$ rounds, the MW Boosting algorithm will output a hypothesis h with error at most ε.*

Main Theorem

Theorem

Let $\varepsilon \in (0,1)$ and $\gamma \in (0, \frac{1}{2})$. Let $\mathcal{F}$ be a class of Boolean functions and let S be a dataset labeled by some $f \in \mathcal{F}$. Given access to a γ-weak learner WkL and run for $T \geq \frac{2\log(1/\varepsilon)}{\gamma^2}$ rounds, the MW Boosting algorithm will output a hypothesis h with error at most ε.

Proof idea (much like last time!): We will bound the total weight W^{T+1} at the end of the algorithm in two ways:

1. A **lower bound** based on the number of mistakes the final hypothesis h makes.
2. An **upper bound** based on the progress made by the weak learner at each step.

Combining these will give us the desired bound on the error of h.

Proof Sketch: Lower Bound

Let $B = \{i \in [m] : h(x^i) \neq y^i\}$ be the set of examples misclassified by the final hypothesis h.

- Since h is a majority vote, for any $i \in B$, at least $T/2$ of the weak hypotheses h_t must have misclassified x^i.

Proof Sketch: Lower Bound

Let $B = \{i \in [m] : h(x^i) \neq y^i\}$ be the set of examples misclassified by the final hypothesis h.

- Since h is a majority vote, for any $i \in B$, at least $T/2$ of the weak hypotheses h_t must have misclassified x^i.
- This means for each $i \in B$, the loss $\ell_i^t = 1$ for at most $T/2$ rounds.

Proof Sketch: Lower Bound

Let $B = \{i \in [m] : h(x^i) \neq y^i\}$ be the set of examples misclassified by the final hypothesis h.

- Since h is a majority vote, for any $i \in B$, at least $T/2$ of the weak hypotheses h_t must have misclassified x^i.
- This means for each $i \in B$, the loss $\ell_i^t = 1$ for at most $T/2$ rounds.
- The final weight on a misclassified example $i \in B$ is:

$$w_i^{T+1} = w_i^1 \prod_{t=1}^{T} e^{-\gamma \ell_i^t} = \exp(-\gamma \sum_{t=1}^{T} \ell_i^t) \geq \exp(-\gamma T/2)$$

Proof Sketch: Lower Bound

Let $B = \{i \in [m] : h(x^i) \neq y^i\}$ be the set of examples misclassified by the final hypothesis h.

- Since h is a majority vote, for any $i \in B$, at least $T/2$ of the weak hypotheses h_t must have misclassified x^i.
- This means for each $i \in B$, the loss $\ell_i^t = 1$ for at most $T/2$ rounds.
- The final weight on a misclassified example $i \in B$ is:

$$w_i^{T+1} = w_i^1 \prod_{t=1}^{T} e^{-\gamma \ell_i^t} = \exp(-\gamma \sum_{t=1}^{T} \ell_i^t) \geq \exp(-\gamma T/2)$$

- The total weight is lower bounded by the sum of weights on these "bad" examples:

$$W^{T+1} = \sum_{i=1}^{m} w_i^{T+1} \geq \sum_{i \in B} w_i^{T+1} \geq |B| e^{-\gamma T/2}$$

Proof Sketch: Upper Bound

We can relate the total weight at step $t+1$ to the total weight at step t:

$$W^{t+1} = \sum_{i \in [m]} w_i^{t+1} = \sum_{i \in [m]} w_i^t e^{-\gamma \ell_i^t}$$

$$= W^t \sum_{i \in [m]} \frac{w_i^t}{W^t} e^{-\gamma \ell_i^t} = W^t \mathop{\mathbb{E}}_{i \sim D^t} [e^{-\gamma \ell_i^t}]$$

Proof Sketch: Upper Bound

We can relate the total weight at step $t+1$ to the total weight at step t:

$$W^{t+1} = \sum_{i \in [m]} w_i^{t+1} = \sum_{i \in [m]} w_i^t e^{-\gamma \ell_i^t}$$

$$= W^t \sum_{i \in [m]} \frac{w_i^t}{W^t} e^{-\gamma \ell_i^t} = W^t \mathop{\mathbb{E}}_{i \sim D^t} [e^{-\gamma \ell_i^t}]$$

By the weak learner guarantee, the expected loss $\mathbf{Pr}_{i \sim D^t}[\ell_i^t = 1]$ is at least $\frac{1}{2} + \gamma$. Using this, we can show that the weight shrinks multiplicatively each round (see notes for proof):

$$W^{t+1} \leq W^t \exp(-\frac{\gamma}{2} - \frac{\gamma^2}{2})$$

Proof Sketch: Upper Bound

We can relate the total weight at step $t+1$ to the total weight at step t:

$$W^{t+1} = \sum_{i \in [m]} w_i^{t+1} = \sum_{i \in [m]} w_i^t e^{-\gamma \ell_i^t}$$

$$= W^t \sum_{i \in [m]} \frac{w_i^t}{W^t} e^{-\gamma \ell_i^t} = W^t \mathop{\mathbb{E}}_{i \sim D^t} [e^{-\gamma \ell_i^t}]$$

By the weak learner guarantee, the expected loss $\mathbf{Pr}_{i \sim D^t}[\ell_i^t = 1]$ is at least $\frac{1}{2} + \gamma$. Using this, we can show that the weight shrinks multiplicatively each round (see notes for proof):

$$W^{t+1} \leq W^t \exp(-\frac{\gamma}{2} - \frac{\gamma^2}{2})$$

After T rounds, with $W^1 = m$:

$$W^{T+1} \leq m \cdot \exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$$

Proof Sketch: Putting It Together

We have just shown upper and lower bounds on W^{T+1}:

- Lower Bound: $|B|\exp(-\frac{\gamma T}{2})$
- Upper Bound: $m\exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$

Proof Sketch: Putting It Together

We have just shown upper and lower bounds on W^{T+1}:

- Lower Bound: $|B|\exp(-\frac{\gamma T}{2})$
- Upper Bound: $m\exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$

Putting everything together:

$$|B|\exp(-\gamma T/2) \leq W^{T+1} \leq m\exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$$

$$\Rightarrow \frac{|B|}{m} \leq \exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2}) \cdot \exp(\gamma T/2) \leq \exp(-\frac{\gamma^2 T}{2})$$

Proof Sketch: Putting It Together

We have just shown upper and lower bounds on W^{T+1}:

- Lower Bound: $|B|\exp(-\frac{\gamma T}{2})$
- Upper Bound: $m\exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$

Putting everything together:

$$|B|\exp(-\gamma T/2) \leq W^{T+1} \leq m\exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2})$$

$$\Rightarrow \frac{|B|}{m} \leq \exp(-\frac{\gamma T}{2} - \frac{\gamma^2 T}{2}) \cdot \exp(\gamma T/2) \leq \exp(-\frac{\gamma^2 T}{2}) \leq \varepsilon$$

To get error $\frac{|B|}{m} \leq \varepsilon$, we want $\exp(-\frac{\gamma^2 T}{2}) \leq \varepsilon$, which means:

$$T \geq \frac{2\log(1/\varepsilon)}{\gamma^2}$$

Other Applications of Multiplicative Weights

Boosting is just one of many applications of the multiplicative weights algorithm and the principle of no-regret learning. Other areas include:

- Approximately solving linear programs
- Game theory (computing approximate Nash equilibria)
- Hardness amplification (constructing strongly secure encryption from weak encryption)
- Network congestion control
- Computational geometry