Introduction to Online Learning Algorithms

Yoav Freund

January 7, 2025

Outline

About this Course

Halving Algorithm

Perceptron

Estimating the mean

Class web site

- All of the class material is available from the github repository https://github.com/yoavfreund/2025-online-learning
-
- Instructor: Yoav Freund: yfreund@ucsd.edu
- TA: Parsa Mirtaheri: smirtaheri@ucsd.edu
- Office Hours: TBD

HW / Evaluation

- 5 HW assignments for 5*15 = 75 opints
- A final for 25 points.

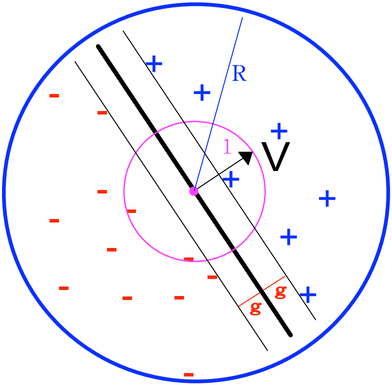
Example trace for Halving Algorithm

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
expert1	1	1	1	1	-
expert2	1	0	-	-	-
expert3	0	-	-	-	-
expert4	1	0	-	-	-
expert5	1	0	-	-	-
expert6	0	-	-	-	-
expert7	1	1	1	1	-
expert8	1	1	1	0	0
alg.	1	0	1	1	0
outcome	1	1	1	0	0

Mistake bound for Halving algorithm

- Each time algorithm makes a mistakes, the pool of perfect experts is halved (at least).
- We assume that at least one expert is perfect.
- Number of mistakes is at most $\log_2 N$.
- No stochastic assumptions whatsoever.
- Proof is based on combining a lower and upper bounds on the number of perfect experts.

The Perceptron Problem

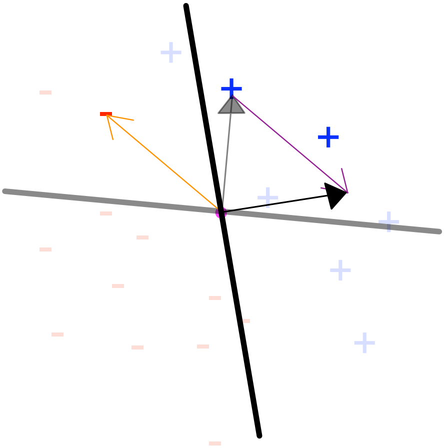


- $\|\vec{V}\| = 1$
- Example = $(\vec{X}, y)$, $y \in \{-1, +1\}$.
- $\forall \vec{X}, \ \ \|\vec{X}\| \leq R$.
- $\forall (\vec{X}, y)$, $y(\vec{X} \cdot \vec{V}) \geq g$

The Perceptron learning algorithm

- An online algorithm. Examples presented one by one.
- start with $\vec{W}_0 = \vec{0}$.
- If mistake: $(\vec{W}_i \cdot \vec{X}_i)y_i \leq 0$
  - Update $\vec{W}_{i+1} = \vec{W}_i + y_i X_i$.

Example trace for the perceptron algorithm



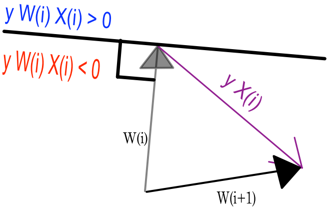
Bound on number of mistakes

- The number of mistakes that the perceptron algorithm can make is at most $\left(\frac{R}{g}\right)^2$.
- Proof by combining upper and lower bounds on $\|\vec{W}\|$.

Pythagorian Lemma

If $(\vec{W}_i \cdot X_i)y < 0$ then

$$\|\vec{W}_{i+1}\|^2 = \|\vec{W}_i + y_i\vec{X}_i\|^2 \leq \|\vec{W}_i\|^2 + \|\vec{X}_i\|^2$$



Upper bound on $\|\vec{W}_i\|$

Proof by induction

- Claim: $\|\vec{W}_i\|^2 \leq iR^2$
- Base: $i = 0$, $\|\vec{W}_0\|^2 = 0$
- Induction step (assume for i and prove for $i + 1$):
  $\|\vec{W}_{i+1}\|^2 \leq \|\vec{W}_i\|^2 + \|\vec{X}_i\|^2$
  $\leq \|\vec{W}_i\|^2 + R^2 \leq (i + 1)R^2$

Lower bound on $\|\vec{W}_i\|$

$\|\vec{W}_i\| \geq \vec{W}_i \cdot \vec{V}$ because $\|\vec{V}\| = 1$.
Let i denote the number of mistakes made so far.

We prove a lower bound on $\vec{W}_i \cdot \vec{V}$ by induction over i

- Claim: $\vec{W}_i \cdot \vec{V} \geq ig$
- Base: $i = 0$, $\vec{W}_0 \cdot \vec{V} = 0$
- Induction step (assume for i and prove for $i + 1$):
  $\vec{W}_{i+1} \cdot \vec{V} = \left(\vec{W}_i + \vec{X}_i y_i\right) \vec{V} = \vec{W}_i \cdot \vec{V} + y_i \vec{X}_i \cdot \vec{V}$
  $\geq ig + g = (i + 1)g$

Combining the upper and lower bounds

$$(ig)^2 \leq \|\vec{W}_i\|^2 \leq iR^2$$

Thus:

$$i \leq \left(\frac{R}{g}\right)^2$$

The mean estimation game

- An adversary choses a real number $y_t \in [0, 1]$ and keeps it secret.
- You make a guess of the secret number x_t
- The adversary reveals the secret and you pay $(x_t - y_t)^2$
- You want to minimize $\frac{1}{T}\sum_{t=1}^{T}(x_t - y_t)^2$
- Impossible without additional constraints.

Adversary is a fixed distribution

- Suppose that the adversary draws $y_1, y_2, \ldots, y_T$ IID from a fixed distribution over $[0, 1]$ with mean μ and std σ.
- Optimal prediction $x_t = \mu$
- $\mathrm{E}_Y \left[(\mu - Y)^2\right] = \sigma^2$
- Online prediction: predict x_{t+1} from $Y^t = \langle Y_1, Y_2, \ldots, Y_t \rangle$.
- **Expected regret**: compare performance of algorithm to Regret $= \mathrm{E}_{Y^T} \left[(x_t - Y_t)^2\right] - \sigma^2$

Individual sequence bounds

- Make no assumption about how the sequence is generated.
- The best constant value for x in hind-sight:

$$x_T^* \doteq \operatorname*{argmin}_{x \in [0,1]} \sum_{t=1}^{T} (x - y_t)^2, \quad x_T^* = \frac{1}{T} \sum_{t=1}^{T} y_t$$

- Regret: the loss over and above the loss of x_T^*. **for the worst-case sequence**
  $\text{Regret}_T = \sum_{t=1}^{T} (x_t - y_t)^2 - \sum_{t=1}^{T} (x_T^* - y_t)^2$
- **Goal:** sublinear regret $\lim_{T \to \infty} \frac{\text{Regret}_T}{T} = 0$

Follow the Leader

- Idea: set x_{t+1} to be the best constant prediction on $y_1, \dots, y_t$
- $x_{t+1} = \operatorname{argmin}_{x \in [0,1]} \sum_{i=1}^{t} (x - y_i)^2 = x_t^*$
- We will prove that the regret of this algorithm is upper bound by $2 + 2 \ln T$

Regret Bound

Theorem

Let $y_t \in [0, 1]$ for $t = 1, \dots T$ an arbitrary sequence of numbers. Let the algorithm output be $x_t = x^*_{t-1} = \frac{1}{t-1} \sum_{i=1}^{t-1} y_i$, then

$$Regret_T = \sum_{t=1}^{T} (x_t - y_t)^2 - \sum_{t=1}^{T} (x^*_T - y_t)^2 \leq 2\,(1 + \ln T)$$

Lemma

Let $x_1, x_2, \ldots$ be the squence of predictions produced by FTL. Then for all $u \in R$ (In particular, for $u = x_T^*$) :

$$\sum_{t=1}^{T} \left((x_t - y_t)^2 - (u - y_t)^2 \right) \leq \sum_{t=1}^{T} \left((x_t - y_t)^2 - (x_t^* - y_t)^2 \right)$$

Proof Sketch:
Subtract $\sum_{t=1}^{T}(x_t - y_t)^2$ from both sides to get an equivalent claim:

$$\sum_{t=1}^{T}(x_t^* - y_t)^2 \leq \sum_{t=1}^{T}(u - y_t)^2$$

The inequality is proven by induction on T.

Proof.

- Base case ($T=1$): $(x_1^* - y_1)^2 = (y_1 - y_1)^2 = 0 \leq (u - y_1)^2$
- Induction hypothesis: $\sum_{t=1}^{T-1}(x_t^* - y_t)^2 \leq \sum_{t=1}^{T-1}(u - y_t)^2$
- Induction step:

$$\sum_{t=1}^{T-1}(x_t^* - y_t)^2 \leq \sum_{t=1}^{T-1}(x_{T-1}^* - y_t)^2 \leq \sum_{t=1}^{T-1}(x_T^* - y_t)^2$$

Adding $(x_T^* - y_T)^2$ to both sides gives:

$$\sum_{t=1}^{T}(x_t^* - y_t)^2 \leq \sum_{t=1}^{T}(x_T^* - y_t)^2 \leq \sum_{t=1}^{T}(u - y_t)^2$$

$\square$

Proof of the theorem

First, note that in FTL we have:

$$x_t = x^*_{t-1} = \frac{1}{t-1}\sum_{i=1}^{t-1} y_i = \frac{t}{t-1}\cdot\left(\frac{1}{t}\sum_{i=1}^{t} y_i - \frac{y_t}{t}\right) = \frac{t}{t-1}\cdot\left(x^*_t - \frac{y_t}{t}\right)$$

Subtracting x^*_t from both sides, we get $x_t - x^*_t = \frac{x^*_t - y_t}{t-1}$. Then:

$$\begin{aligned}
\text{Regret}_T &= \sum_{t=1}^{T}(x_t - y_t)^2 - \sum_{t=1}^{T}(x^*_T - y_t)^2 \\
&\le \sum_{t=1}^{T}(x_t - y_t)^2 - (x^*_t - y_t)^2 \qquad \text{(Lemma)} \\
&= \sum_{t=1}^{T}(x_t + x^*_t - 2y_t)(x_t - x^*_t) \le \sum_{t=1}^{T}\frac{2}{t-1} \le 2\,(1 + \ln T)
\end{aligned}$$

What the class will cover

- Introduction (with Mean)
- Exponential weights algorithms
  - Hedge
  - Mixability
  - Bregman Divergences
- Online learning and Coding
  - Universal Coding
  - Continuous Experts
  - The Context Algorithm
- Multiple arm Bandit
- Tracking
  - Tracking
  - Tracking within a small set of experts
- Online learning and game theory
  - Reepeated Matrix Games
  - Internal regret.
  - Drifting games
  - NormalHedge
- Online Convex Optimizatio
  - Follow the regularized leader
  - Dual Descent
  - AdaGrad