Introduction to Online Learning Algorithms

Yoav Freund

January 9, 2018

Outline

Halving Algorithm

Hedge Algorithm

Perceptron

Laplace law of succession

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	0
expert8	1	1	1	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 hedging problem

- N possible actions
- At each time step t:
  - Algorithm chooses a distribution $\vec{P}^t$ over actions.
  - Losses $0 \leq \ell_i^t \leq 1$ of all actions $i = 1, \ldots, N$ are revealed.
  - Algorithm suffers **expected** loss.
- **Goal:** minimize total expected loss
- Here we have stochasticity - but only in **algorithm**, not in **outcome**
- Fits nicely in game theory

Hedging vs. Halving

- Like halving - we want to zoom into best action (expert).
- Unlike halving - no action is perfect.
- Basic idea - reduce probability of lossy actions,
  but not all the way to zero.
- **Modified Goal:** minimize difference between
  expected total loss
  and
  minimal total loss of repeating one action.

The Hedge Algorithm

Consider action i at time t

- Total loss:
$$L_i^t = \sum_{s=1}^{t-1} \ell_i^s$$
- Weight:
$$W_i^t = e^{-\eta L_i^t}$$
- $\eta > 0$ is the learning rate parameter. Halving: $\eta = \infty$
- Probability:
$$P_i^t = \frac{W_i^t}{\sum_{j=1}^{N} W_j^t}$$

Example trace for Hedge Algorithm

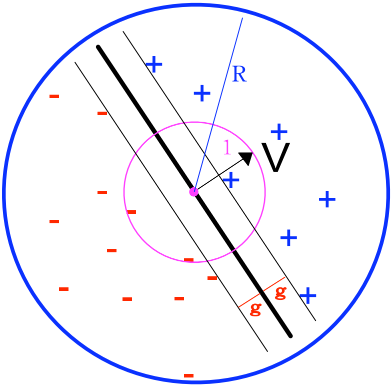
$\eta = 1$

	$\vec{W}^1$	$\vec{\ell}^1$	$\vec{W}^2$	$\vec{\ell}^2$	$\vec{W}^3$	$\vec{\ell}^3$	total
expert1	1	.1	.90	.1	0.82	0	.2
expert2	1	.8	.45	.5	0.27	.2	1.5
expert3	1	.3	.74	.2	0.61	.2	.7
expert4	1	.1	.90	.7	0.45	.8	1.6
expert5	1	.9	.41	1	0.15	.8	2.7
expert6	1	0	1	.1	0.91	.2	.3
expert7	1	1	.37	.5	0.22	.4	1.9
expert8	1	.8	.45	.2	0.37	.6	1.6
alg.		.5		.36		.30	1.16

Bound for Hedge Algorithm

- L^t_{Hedge}: Expected total loss of Hedge algorithm for time $1, 2, \ldots, t$
- $$\forall t, i, \quad L_{\text{Hedge}} \le \frac{\ln N + \eta L_i^t}{1 - e^{-\eta}}$$
- Which implies
$$\forall t, \quad L_{\text{Hedge}} \le \min_i \left(\frac{\ln N + \eta L_i^t}{1 - e^{-\eta}} \right)$$
- Proof and choice of η: next class.

The Perceptron Problem

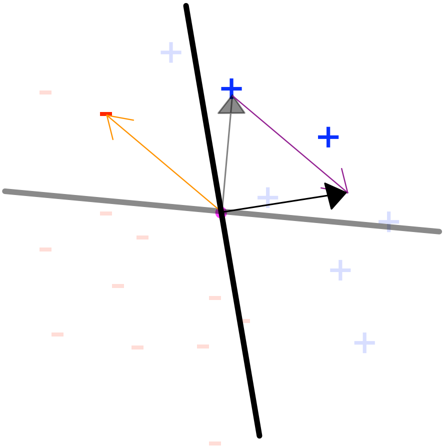


- $\|\vec{V}\| = 1$
- Example = $(\vec{X}, y)$, $y \in \{-1, +1\}$.
- $\forall \vec{X},\ \ \|\vec{X}\| \leq R$.
- $\forall (\vec{X}, y)$, $y(\vec{X} \cdot \vec{V}) \geq g$

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



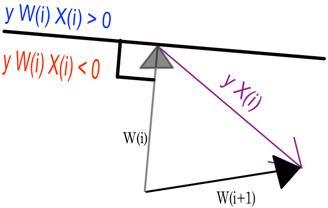
Bound on number of mistakes

- 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 $\|\vec{V}\| = 1$.
We prove a lower bound on $\vec{W}_i \cdot \vec{V}$ using 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:

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

Estimating the bias of a coin

- We observe n coin flips:
  H,T,T,H,H,T,H,T,T
- We want to estimate the **probability** that the next flip will be **H**ead.
- Natural Answer:

$$\frac{\#\mathbf{H}}{n} = \frac{4}{9}$$

What if the estimation has to be done online?

- We observe a bf sequence of coin flips, and have to predict the probability of **H**ead at each step:
  p_0,**H**,p_1,**T**,p_2,...
- What would be a good value for p_0?
- For p_1?
- Laplace Law of succession
$$\frac{\#\mathbf{H}+1}{n+2}$$
- Turns out that a better rule is
$$\frac{\#\mathbf{H}+1/2}{n+1}$$
  Krichevsky and Trofimov, 1981
- Why?
- What does “better” mean?

To be continued...

Please

- Register on twiki (follow directions on my home page)
- Follow link from main twiki page to “Online learning course”
- Add yourself to the list and the table on `ClassParticipants`
- Go to `CoursePlan/LessonNo1` to see slides and to post questions and answers.
- See you on Thursday!