Online Algorithms:
Learning & Optimization with No Regret.

CS/CNS/EE 253
Daniel Golovin
The Setup

Optimization:
- Model the problem (objective, constraints)
- Pick best decision from a feasible set.

Learning:
- Model the problem (objective, hypothesis class)
- Pick best hypothesis from a feasible set.
Online Learning/Optimization

Choose an action

\[ x_t \in X \]

Get \( f_t(x_t) \) and feedback

\[ f_t : X \to [0, 1] \]

- Same feasible set \( X \) in each round \( t \)
- Different Reward Models:
  - Stochastic, Arbitrary but Oblivious, Adaptive and Arbitrary
Concrete Example: Commuting

Pick a path $x_t$ from home to school.
Pay cost $f_t(x_t) := \sum_{e \in x_t} c_t(e)$
Then see all edge costs for that round.

Dealing with Limited Feedback: later in the course.
Other Applications

- Sequential decision problems
- Streaming algorithms for optimization/learning with large data sets
- Combining weak learners into strong ones ("boosting")
- Fast approximate solvers for certain classes of convex programs
- Playing repeated games
Binary prediction with a perfect expert

- $n$ hypotheses ("experts") $h_1, h_2, \ldots, h_n$
- Guaranteed that some hypothesis is perfect.
- Each round, get a data point $p_t$ and classifications $h_i(p_t) \in \{0, 1\}$
- Output binary prediction $x_t$, observe correct label
- Minimize # mistakes

Any Suggestions?
A Weighted Majority Algorithm

- Each expert “votes” for it's classification.
- Only votes from experts who have never been wrong are counted.
- Go with the majority

\[ \# \text{ mistakes } M \leq \log_2(n) \]

Weights \( w_{it} = I(h_i \text{ correct on first } t \text{ rounds}) \).
\[ W_t = \sum_i w_{it} \]
\[ W_0 = n, \quad W_T \geq 1 \]
Mistake on round \( t \) implies \( W_{t+1} \leq W_t/2 \)
So \( 1 \leq W_T \leq W_0/2^M = n/2^M \)
Weighted Majority

[Littlestone & Warmuth '89]

What if there's no perfect expert?

- Each expert $i$ has a weight $w(i)$, “votes” for it's classification in $\{-1, 1\}$.

Go with the weighted majority, predict $\text{sign}(\sum_i w_i x_i)$.

Halve weights of wrong experts. Let $m = \#$ mistakes of best expert. How many mistakes $M$ do we make?

Weights $w_{it} = (1/2)^{\# \text{ mistakes by } i \text{ on first } t \text{ rounds}}$.

Let $W_t := \sum_i w_{it}$.

Note $W_0 = n$, $W_T \geq (1/2)^m$

Mistake on round $t$ implies $W_{t+1} \leq \frac{3}{4} W_t$

So $(1/2)^m \leq W_T \leq W_0 (3/4)^M = n \cdot (3/4)^M$

Thus $(4/3)^M \leq n \cdot 2^m$ and $M \leq 2.41(m + \log_2(n))$. 
Can we do better?

\[ M \leq 2.41(m + \log_2(n)) \]

<table>
<thead>
<tr>
<th>Experts</th>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>e_1 \equiv -1</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>e_2 \equiv 1</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

• No deterministic algorithm can get \( M < 2m \).
• What if there are more than 2 choices?
Regret

“Maybe all one can do is hope to end up with the right regrets.” – Arthur Miller

- Notation: Define loss or cost functions $c_t$ and define the regret of $x_1, x_2, \ldots, x_T$ as

$$R_T = \sum_{t=1}^{T} c_t(x_t) - \sum_{t=1}^{T} c_t(x^*)$$

where $x^* = \arg\min_{x \in X} \sum_{t=1}^{T} c_t(x)$

A sequence has “no-regret” if $R_T = o(T)$.

- Questions:
  - How can we improve Weighted Majority?
  - What is the lowest regret we can hope for?
The Hedge/WMR Algorithm*

[Hasek & Schapire '97]

Hedge(ε)
Initialize \( w_{i0} = 1 \) for all \( i \).
In each round \( t \):
Choose expert \( e_t \) from categorical distribution \( p_t \)
Select \( x_t = x(e_t, t) \), the advice/prediction of \( e_t \).
For each \( i \), set \( w_{i,t+1} = w_{it}(1 - \epsilon)^{c_t(x(e_i,t))} \)

• How does this compare to WM?

* Pedantic note: Hedge is often called “Randomized Weighted Majority”, and abbreviated “WMR”, though WMR was published in the context of binary classification, unlike Hedge.
The Hedge/WMR Algorithm

Hedge(\(\epsilon\))
Initialize \(w_{i0} = 1\) for all \(i\).
In each round \(t\):

Choose expert \(e_t\) from categorical distribution \(p_t\)
Select \(x_t = x(e_t, t)\), the advice/prediction of \(e_t\).
For each \(i\), set \(w_{i,t+1} = w_{it} (1 - \epsilon)^{c_t(x(e_i, t))}\)

Randomization

Influence shrinks exponentially with cumulative loss.
Intuitively: Either we do well on a round, or total weight drops, and total weight can't drop too much unless every expert is lousy.
Hedge Performance

Theorem: Let $x_1, x_2, \ldots$ be the choices of Hedge($\epsilon$). Then

$$\mathbb{E} \left[ \sum_{t=1}^{T} c_t(x_t) \right] \leq \left( \frac{1}{1 - \epsilon} \right) \text{OPT}_T + \frac{\ln(n)}{\epsilon}$$

where $\text{OPT}_T := \min_i \sum_{t=1}^{T} c_t(x(e_i, t))$.

If $\epsilon = \Theta \left( \sqrt{\ln(n)/\text{OPT}} \right)$, the regret is $\Theta(\sqrt{\text{OPT} \ln(n)})$
Hedge Analysis

Intuitively: Either we do well on a round, or total weight drops, and total weight can't drop too much unless every expert is lousy.

Let $W_t := \sum_i w_{it}$. Then $W_0 = n$ and $W_{T+1} \geq (1 - \epsilon)^{0\text{PT}}$.

$$W_{t+1} = \sum_i w_{it} (1 - \epsilon)^{c_t(x_{it})}$$

$$= \sum_i W_t p_t(i) (1 - \epsilon)^{c_t(x_{it})}$$

[def of $p_t(i)$] (2)

$$\leq \sum_i W_t p_t(i) (1 - \epsilon \cdot c_t(x_{it}))$$ [Bernoulli's ineq]

$$= W_t (1 - \epsilon \cdot \mathbb{E} [c_t(x_t)])$$

$$\leq W_t \cdot \exp (-\epsilon \cdot \mathbb{E} [c_t(x_t)])$$ [1 - $x \leq e^{-x}$] (5)
Hedge Analysis

\[ W_{T+1}/W_0 \leq \exp \left( -\epsilon \sum_{t=1}^{T} \mathbb{E} [c_t(x_t)] \right) \]

\[ W_0/W_{T+1} \geq \exp \left( \epsilon \sum_{t=1}^{T} \mathbb{E} [c_t(x_t)] \right) \]

Recall \( W_0 = n \) and \( W_{T+1} \geq (1 - \epsilon)^{OPT} \).

\[ \mathbb{E} \left[ \sum_{t=1}^{T} c_t(x_t) \right] \leq \frac{1}{\epsilon} \ln \left( \frac{W_0}{W_{T+1}} \right) \leq \frac{\ln(n)}{\epsilon} - \frac{OPT \cdot \ln(1 - \epsilon)}{\epsilon} \]

\[ \leq \frac{\ln(n)}{\epsilon} + \frac{OPT}{1 - \epsilon} \]
Lower Bound

If $\epsilon = \Theta\left(\frac{\sqrt{\ln(n)}}{OPT}\right)$, the regret is $\Theta\left(\sqrt{OPT \ln(n)}\right)$

Can we do better?

Let $c_t(x) \sim \text{Bernoulli}(1/2)$ for all $x$ and $t$.
Let $Z_i := \sum_{t=1}^{T} c_t(x(e_i, t))$.
Then $Z_i \sim \text{Bin}(T, 1/2)$ is roughly normally distributed, with $\sigma = \frac{1}{2} \sqrt{T}$.

$\mathbb{P}[Z_i \leq \mu - k\sigma] = \exp\left(-\Theta(k^2)\right)$

We get about $\mu = T/2$, best choice is likely to get $\mu - \Theta(\sqrt{T \ln(n)}) = \mu - \Theta(\sqrt{OPT \ln(n)})$. 
What have we shown?

- Simple algorithm that learns to do nearly as well as best fixed choice.
  - Hedge can exploit any pattern that the best choice does.

- Works for Adaptive Adversaries.
  - Suitable for playing repeated games. Related ideas appearing in Algorithmic Game Theory literature.
Related Questions

- Optimize and get no-regret against richer classes of strategies/experts:
  - All distributions over experts
  - All sequences of experts that have K transitions [Auer et al '02]
  - Various classes of functions of input features [Blum & Mansour '05]
    - E.g., consider time of day when choosing driving route.
  - Arbitrary convex set of experts, metric space of experts, etc, with linear, convex, or Lipschitz costs. [Zinkevich '03, Kleinberg et al '08]
  - All policies of a K-state initially unknown Markov Decision Process that models the world. [Auer et al '08]
  - Arbitrary sets of strategies in $\mathbb{R}^n$ with linear costs that we can optimize offline. [Hannan'57, Kalai & Vempala '02]
Related Questions

• Other notions of regret (see e.g., [Blum & Mansour '05])
  • Time selection functions:
    - get low regret on mondays, rainy days, etc.
  • Sleeping experts:
    - if rule “if(P) then predict Q” is right 90% of the time it applies, be right 89% of the time P applies.
  • Internal regret & swap regret:
    - If you played $x_1, \ldots, x_T$ then have no regret against $g(x_1), \ldots, g(x_T)$ for every $g:X \rightarrow X$
Sleeping Experts

[Freund et al '97, Blum '97, Blum & Mansour '05]

- if rule “if(P) then predict Q” is right 90% of the time it applies, be right 89% of the time P applies. **Get this for every rule simultaneously.**

- Idea: Generate lots of hypotheses that “specialize” on certain inputs, some good, some lousy, and combine them into a great classifier.

- Many applications:
  - Document classification, Spam filtering, Adaptive Uis, ...
  - if (“physics” in D) then classify D as “science”.

- **Predicates can overlap.**
Sleeping Experts

• Predicates can overlap
  • E.g., predict college major given the classes C you're enrolled in?
    – if(ML-101, CS-201 in C) then CS
    – if(ML-101, Stats-201 in C) then Stats
  • What do we predict for students enrolled in ML-101, CS-201, and Stats-201?
Sleeping Experts
[Algorithm from Blum & Mansour '05]

SleepingExperts(β, E, F)
Input: β ∈ (0, 1), experts E, time selection functions F
Initialize \( w^0_{e,f} = 1 \) for all \( e ∈ E, f ∈ F \).

In each round \( t \):
- Let \( w^t_e = \sum_f f(t) w^t_{e,f} \).
- Let \( W^t = \sum_e w^t_e \).
- Let \( p^t_e = w^t_e / W^t \).

Choose expert \( e_t \) from categorical distribution \( p^t \)
Select \( x_t = x(e_t, t) \), the advice/prediction of \( e_t \).

For each \( e ∈ E, f ∈ F \)
\[
\begin{align*}
    w^{t+1}_{e,f} &= w^t_{e,f} \beta f(t) (c_t(e) - β \mathbb{E}[c_t(e_t)])
\end{align*}
\]
Sleeping Experts
[Algorithm from Blum & Mansour '05]

\[ w_{e,f}^{t+1} = w_{e,f}^t \beta f(t)(c_t(e) - \beta \mathbb{E}[c_t(e_t)]) \]

Ensures total sum of weights can never increase.

\[ \sum_{e,f} w_{e,f}^t \leq nm \text{ for all } t \]

\[ w_{e,f}^T = \prod_{t \geq 0} \beta f(t)(c_t(e) - \beta \mathbb{E}[c_t(e_t)]) \]

\[ = \beta \sum_{t \geq 0} [f(t)(c_t(e) - \beta \mathbb{E}[c_t(e_t)])] \]

\[ \leq nm \]
Sleeping Experts Performance

Let \( n = |\mathcal{E}|, m = |\mathcal{F}|. \) Fix \( T \in \mathbb{N}. \)

Let \( C(e, f) := \sum_{t=1}^{T} f(t) \cdot c_t(e) \)

Let \( C_{\text{alg}}(f) := \sum_{t=1}^{T} f(t) \cdot c_t(e_t) \)

Then for all \( e \in \mathcal{E}, f \in \mathcal{F} \)

\[
\mathbb{E} \left[ C_{\text{alg}}(f) \right] \leq \frac{1}{\beta} \left( C(e, f) + \log_{1/\beta}(nm) \right)
\]

If \( \beta = 1 - \epsilon \) is close to 1,

\[
\mathbb{E} \left[ C_{\text{alg}}(f) \right] = (1 + \Theta(\epsilon)) C(e, f) + \Theta \left( \frac{\log_2(nm)}{\epsilon} \right)
\]

Optimizing yields a regret bound of

\[
O\left( \sqrt{C(e, f) \log(nm)} + \log(nm) \right).
\]