Prediction, Learning and Games - Chapter 5

Jérôme Thai

UC Berkeley

October 21, 2013
Recall

- On iteration $t$: experts reveal their advice "i"
Recall

- On iteration $t$: experts reveal their advice "i"
- forecaster makes a decision $\hat{p}_t = \sum_{i=1}^{N} w_{i,t}$
EXPERTS FRAMEWORK

Recall

- On iteration $t$: experts reveal their advice "i"
- forecaster makes a decision $\hat{p}_t = \sum_{i=1}^{N} w_{i,t} i$
- the losses are revealed $\ell(i, y_t)$ and $\ell(\hat{p}_t, y_t)$
Recall

- On iteration $t$: experts reveal their advice "i"
- forecaster makes a decision $\hat{p}_t = \sum_{i=1}^{N} w_{i,t}$
- the losses are revealed $\ell(i, y_t)$ and $\ell(\hat{p}_t, y_t)$
- forecaster updates weights $w_{i,t+1}$
Recall

- On iteration $t$: experts reveal their advice "$i$"
- forecaster makes a decision $\hat{p}_t = \sum_{i=1}^{N} w_{i,t} i$
- the losses are revealed $\ell(i, y_t)$ and $\ell(\hat{p}_t, y_t)$
- forecaster updates weights $w_{i,t+1}$
- $\ell(\cdot, y_t)$ is convex
Experts framework

Recall

- On iteration $t$: experts reveal their advice “$i$”
- forecaster makes a decision $\hat{p}_t = \sum_{i=1}^{N} w_{i,t} i$
- the losses are revealed $\ell(i, y_t)$ and $\ell(\hat{p}_t, y_t)$
- forecaster updates weights $w_{i,t+1}$
- $\ell(\cdot, y_t)$ is convex

Regret: $R_{i,T} = \hat{L}_T - L_{i,T} = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \ell(i, y_t)$

Goal: $\frac{R_T}{T} = o(T)$
MULTIPlicative weight algorithms

Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \)
Multiplicative weight algorithms

Hedge algorithm: $w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t))$

- Regret $R_T \leq \frac{\ln N}{\gamma} + \frac{\gamma T}{8}$
MULTIPlicative weight algorithms

Hedge algorithm: \[ w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \]

- Regret \( R_T \leq \frac{\ln N}{\gamma} + \frac{\gamma T}{8} \)
- Taking \( \gamma_t = \sqrt{\frac{8 \ln N}{T}} \) yields \( R_T \leq \sqrt{\frac{T}{2} \ln N} \)
Multiplicative weight algorithms

Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \)

- Regret \( R_T \leq \frac{\ln N}{\gamma} + \frac{\gamma T}{8} \)
- taking \( \gamma_t = \sqrt{\frac{8 \ln N}{T}} \) yields \( R_T \leq \sqrt{\frac{T}{2}} \ln N \)
- small losses: \( \frac{R_T}{T} \leq \frac{1}{T} \left( \frac{\gamma}{1-e^{-\gamma}} - 1 \right) L_i^* + \frac{1}{T} \frac{\ln N}{1-e^{-\gamma}} \)
PROBLEM
PROBLEM

- N experts
**PROBLEM**

- N experts
- Hedge algorithm:  
  
  \[ w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \quad i \in [N] \]
PROBLEM

- $N$ experts
- Hedge algorithm: $w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \quad i \in [N]$
- complexity $\propto N$
PROBLEM

- N experts
- Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \quad i \in [N] \)
- complexity \( \propto N \)
- N very large \( \implies \) Hedge algorithm infeasible
PROBLEM

- N experts
- Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \) \( i \in [N] \)
- complexity \( \propto N \)
- N very large \( \implies \) Hedge algorithm infeasible
- Experts assume a certain structure
PROBLEM

- N experts
- Hedge algorithm: $w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i,y_t)) \quad i \in [N]$
- complexity $\propto N$
- N very large $\implies$ Hedge algorithm infeasible
- Experts assume a certain structure
- possible construction of efficient prediction algorithms
PROBLEM

- $N$ experts
- Hedge algorithm: $w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \quad i \in [N]$
- complexity $\propto N$
- $N$ very large $\implies$ Hedge algorithm infeasible
- Experts assume a certain structure
- possible construction of efficient prediction algorithms
- two cases: *Tracking the best expert* and *tree experts*
PROBLEM

- N experts
- Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \) \( i \in [N] \)
- complexity \( \propto N \)
- N very large \( \implies \) Hedge algorithm infeasible
- Experts assume a certain structure
- possible construction of efficient prediction algorithms
- two cases: Tracking the best expert and tree experts
- Idea: track N “base” experts (actions) for \( M = f(N) \) experts
PROBLEM

- N experts
- Hedge algorithm: \( w_{i,t+1} \propto w_{i,t} \exp(-\gamma \ell(i, y_t)) \) \( i \in [N] \)
- complexity \( \propto N \)
- N very large \( \Rightarrow \) Hedge algorithm infeasible
- Experts assume a certain structure
- possible construction of efficient prediction algorithms
- two cases: Tracking the best expert and tree experts
- Idea: track \( N \) “base” experts (actions) for \( M = f(N) \) experts
- Same weights, same bound \( R_T \leq \sqrt{\frac{T}{2} \ln M} \)
Tracking Best Expert: Setting

Regret against the best performing single action:

\[ R_T = \hat{L}_T - \min_{i=1, \ldots, N} L_{i,T} = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{i=1, \ldots, N} \sum_{t=1}^{T} \ell(i, y_t) \]
Tracking Best Expert: Setting

Regret against the best performing single action:

\[ R_T = \hat{L}_T - \min_{i=1, \ldots, N} L_{i,T} = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{i=1, \ldots, N} \sum_{t=1}^{T} \ell(i, y_t) \]

When allowed to switch actions, Tracking regret:

\[ \tilde{R}_T = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{(i_1, \ldots, i_T)} \sum_{t=1}^{T} \ell(i_t, y_t) \]

where \((i_1, \ldots, i_T) \in \{1, \ldots, N\}^T\)
**Tracking Best Expert: Setting**

Regret against the best performing single action:

$$R_T = \hat{L}_T - \min_{i=1,\ldots,N} L_{i,T} = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{i=1,\ldots,N} \sum_{t=1}^{T} \ell(i, y_t)$$

When allowed to switch actions, Tracking regret:

$$\tilde{R}_T = \sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{(i_1,\ldots,i_T)} \sum_{t=1}^{T} \ell(i_t, y_t)$$

where $$(i_1,\ldots,i_T) \in \{1,\ldots,N\}^T$$

- $N$ "base" experts $i \in \{1,\ldots,N\} = [N]$
- $N^T$ "compound" experts $$(i_1,\ldots,i_T) \in [N]^T$$
**BOUNDED NUMBER OF SWITCHES**

We impose $\leq m$ switches:

$$\text{actions: } \{(i_1, \cdots, i_1, i_2, \cdots, i_2, \cdots, i_{m+1}, \cdots, i_{m+1})\}$$
**BOUNDED NUMBER OF SWITCHES**

We impose $\leq m$ switches:

$$\text{actions: } \{(i_1, \cdots, i_1, \ i_2, \cdots, i_2, \ \cdots, i_{m+1}, \cdots, i_{m+1})\}$$

- less “compound experts”
- more tractable, scalable algorithms
**Bounded number of switches**

We impose \( \leq m \) switches:

\[
\text{actions: } \{(i_1, \ldots, i_1, i_2, \ldots, i_2, \ldots, i_{m+1}, \ldots, i_{m+1})\}
\]

- less “compound experts”
- more tractable, scalable algorithms

\[
M := \# \text{“compound” experts}
\]

\[
= \sum_{k=0}^{m} \binom{T-1}{k} N(N-1)^k \
\leq N^{m+1} \exp \left((T-1) H\left(m/(T-1)\right)\right)
\]

with \( H(x) = -x \ln x - (1 - x) \ln(1 - x) \)
**Bounded Number of Switches**

We impose $\leq m$ switches:

$$\text{actions: } \{(i_1, \cdots, i_1, i_2, \cdots, i_2, \cdots, i_{m+1}, \cdots, i_{m+1})\}$$

- less “compound experts”
- more tractable, scalable algorithms

$$M := \# \text{“compound” experts}$$

$$= \sum_{k=0}^{m} \binom{T-1}{k} N(N - 1)^k$$

$$\leq N^{m+1} \exp \left( (T - 1) H(m/(T - 1)) \right)$$

with $H(x) = -x \ln x - (1 - x) \ln(1 - x)$

$$\tilde{R}_T \leq \sqrt{\frac{T}{2} \ln N} = \sqrt{\frac{T}{2} ((m + 1) \ln M + (T - 1) H(m/(T - 1)))}$$
THE FIXED SHARE FORECASTER

Parameters: Real numbers $\eta > 0$ and $0 \leq \alpha \leq 1$.

Initialization: $w_0 = (1/N, \ldots, 1/N)$.

For each round $t = 1, 2, \ldots$

1. draw an action $I_t$ from $\{1, \ldots, N\}$ according to the distribution

   $$p_{i,t} = \frac{w_{i,t-1}}{\sum_{j=1}^{N} w_{j,t-1}}, \quad i = 1, \ldots, N.$$

2. obtain $Y_t$ and compute

   $$v_{i,t} = w_{i,t-1} e^{-\eta \ell(i, Y_t)} \quad \text{for each } i = 1, \ldots, N.$$

3. let

   $$w_{i,t} = \alpha \frac{W_t}{N} + (1 - \alpha)v_{i,t} \quad \text{for each } i = 1, \ldots, N,$$

   where $W_t = v_{1,t} + \cdots + v_{N,t}$. 
THE FIXED SHARE FORECASTER

Theorem 5.1. Distribution of the action $I_t$ by the fixed share forecaster = distribution of action $I'_t$ by the Hedge algorithm (with specific initialization).
THE FIXED SHARE FORECASTER

Theorem 5.1. Distribution of the action $I_t$ by the fixed share forecaster = distribution of action $I'_t$ by the Hedge algorithm (with specific initialization).

Theorem 5.2. For all compound actions $(i_1, \cdots, i_T)$ with $\leq m$ switches, the tracking regret of the fixed share forecaster satisfies:

$$\tilde{R}_T \leq \frac{m + 1}{\gamma} \ln N + \frac{1}{\gamma} \ln \frac{1}{(\alpha/N)^m (1 - \alpha)^{T-m-1}} + \frac{\gamma T}{8}$$
THE FIXED SHARE FORECASTER

**Theorem 5.1.** Distribution of the action $I_t$ by the fixed share forecaster = distribution of action $I'_t$ by the Hedge algorithm (with specific initialization).

**Theorem 5.2.** For all compound actions $(i_1, \cdots, i_T)$ with $\leq m$ switches, the tracking regret of the fixed share forecaster satisfies:

$$
\tilde{R}_T \leq \frac{m + 1}{\gamma} \ln N + \frac{1}{\gamma} \ln \frac{1}{(\alpha/N)^m(1 - \alpha)^{T-m-1}} + \frac{\gamma}{8} T
$$

**Corollary 5.1.** For $\alpha = m/(T-1)$, and

$$
\sqrt{\frac{8}{T} \left( (m+1) \ln M + (T-1) H(m/(T-1)) \right)}
$$

we have same performance bound as Hedge algorithm.
The fixed share forecaster

Proof of Th 5.1. Let $w^t(i_1, \cdots, i_T)$ = weight of $(i_1, \cdots, i_T)$ at $t$ for Hedge algorithm. Initialize:

$$w_0(i_1, \cdots, i_T) = \frac{1}{N} \left( \frac{\alpha}{N} \right)^{\text{size}(i_1, \cdots, i_T)} \left( 1 - \alpha + \frac{\alpha}{N} \right)^{T-\text{size}(i_1, \cdots, i_T)}$$ (1)
THE FIXED SHARE FORECASTER

**Proof of Th 5.1.** Let $w^t(i_1, \cdots, i_T) = \text{weight of } (i_1, \cdots, i_T) \text{ at } t \text{ for Hedge algorithm. Initialize:}

$$w_0(i_1, \cdots, i_T) = \frac{1}{N} \left( \frac{\alpha}{N} \right)^{\text{size}(i_1, \cdots, i_T)} \left( 1 - \alpha + \frac{\alpha}{N} \right)^{T - \text{size}(i_1, \cdots, i_T)}$$  \hspace{1cm} (1)

Update: $w'_t(i_1, \cdots, i_T) = w'_{t-1}(i_1, \cdots, i_T) \exp (-\gamma \ell(i_t, y_t))$
The fixed share forecaster

**Proof of Th 5.1.** Let $w^t(i_1, \cdots, i_T) =$ weight of $(i_1, \cdots, i_T)$ at $t$ for Hedge algorithm. Initialize:

$$w_0(i_1, \cdots, i_T) = \frac{1}{N} \left( \frac{\alpha}{N} \right)^{\text{size}(i_1, \cdots, i_T)} \left( 1 - \alpha + \frac{\alpha}{N} \right)^{T-\text{size}(i_1, \cdots, i_T)}$$ (1)

Update: $w'_t(i_1, \cdots, i_T) = w'_{t-1}(i_1, \cdots, i_T) \exp(-\gamma \ell(i_t, y_t))$

Choose with distribution $p_{i,t} \propto w'_{i,t} = \sum_{i_1, \cdots, i_T | i_{t+1}=i} w'_t(i_1, \cdots, i_T)$
**The fixed share forecaster**

**Proof of Th 5.1.** Let $w^t(i_1, \cdots, i_T) =$ weight of $(i_1, \cdots, i_T)$ at $t$ for Hedge algorithm. Initialize:

$$w_0(i_1, \cdots, i_T) = \frac{1}{N} \left( \frac{\alpha}{N} \right)^{\text{size}(i_1, \cdots, i_T)} \left( 1 - \alpha + \frac{\alpha}{N} \right)^{T-\text{size}(i_1, \cdots, i_T)}$$  \hspace{1cm} (1)

Update: $w'_t(i_1, \cdots, i_T) = w'_{t-1}(i_1, \cdots, i_T) \exp(-\gamma \ell(i_t, y_t))$

Choose with distribution $p_{i,t} \propto w'_{i,t} = \sum_{i_1, \cdots, i_T | i_{t+1}=i} w'_t(i_1, \cdots, i_T)$

Then we have $w_{i,t} = w'_{i,t}$ by induction.
THE FIXED SHARE FORECASTER

Lemma 5.1. Hedge algorithm with initial weights $w_{1,0} + \cdots + w_{N,0} \leq 1$, then

$$\sum_{t=1}^{T} \ell(\hat{p}_t, y_t) \leq \frac{1}{\gamma} \ln \frac{1}{W_T} + \frac{\gamma T}{8}$$

with $W_T = \sum_{i=1}^{N} w_{i,T} = \sum_{i=1}^{N} w_{i,0} \exp(-\gamma \sum_{t=1}^{T} \ell(i, y_t))$
The fixed share forecaster

**Lemma 5.1.** Hedge algorithm with initial weights $w_{1,0} + \cdots + w_{N,0} \leq 1$, then

$$
\sum_{t=1}^{T} \ell(\hat{p}_t, y_t) \leq \frac{1}{\gamma} \ln \frac{1}{W_T} + \frac{\gamma T}{8}
$$

with $W_T = \sum_{i=1}^{N} w_{i,T} = \sum_{i=1}^{N} w_{i,0} \exp(-\gamma \sum_{t=1}^{T} \ell(i, y_t))$

**Proof of Th 5.2.** Apply lemma 5.1 with weights (1)
Tree Experts: setting

\[ T \]

\[ E \]

\[ N = 4 \text{ actions ("base" experts)}. \]
**Tree Experts: setting**

$N = 4$ actions ("base" experts).

Tree expert $E$

$= \text{binary tree } T \text{ with leaves labeled with actions } \in \{1, 2, 3, 4\}.$
Tree Experts: setting

- $N = 4$ actions ("base" experts).
- Tree expert $E$ = binary tree $T$ with leaves labeled with actions $\in \{1, 2, 3, 4\}$.
- Side information: $x = (x_1, x_2, \cdots) \in \{0, 1\}^\mathbb{N}$.
TREE EXPERTS: setting

$N = 4$ actions ("base" experts).

Tree expert $E$
= binary tree $T$ with leaves labeled with actions $\in \{1, 2, 3, 4\}$.

side information: $x = (x_1, x_2, \cdots) \in \{0, 1\}^\mathbb{N}$

- $x = (0, \cdots) \implies i_E(x) = 2$ "E chooses action 2 given $x$"
- $x = (1, 0, \cdots) \implies i_E(x) = 4$ "E chooses action 4 given $x$"
- $x = (1, 1, \cdots) \implies i_E(x) = 1$ "E chooses action 1 given $x$"
Tree Experts: Setting

Regret against tree expert E:

\[
\overline{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)
\]
**Tree Experts: Setting**

Regret against tree expert $E$:

$$\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)$$

Some definitions:

- $\text{depth}(E) \leq D \implies$ finite # of trees
Tree Experts: Setting

Regret against tree expert $E$:

$$\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)$$

Some definitions:

- depth($E$) $\leq D \implies$ finite # of trees
- length($x_t$) = $D$
Tree Experts: Setting

Regret against tree expert E:

\[ \bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t) \]

Some definitions:

- depth(E) \leq D \implies \text{finite \# of trees}
- length(x_t) = D
- \|E\| := |\{v \in \text{nodes}(E)\}|
**Tree Experts: Setting**

Regret against tree expert $E$:

$$\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)$$

Some definitions:

- $\text{depth}(E) \leq D \implies \text{finite # of trees}$
- $\text{length}(x_t) = D$
- $\| E \| := |\{v \in \text{nodes}(E)\}|$
- $\| E \|_D := \| E \| - |\{v \in \text{leaves}(E) : \text{depth}(v) = D\}|$
**Tree Experts: Setting**

Regret against tree expert $E$:

$$
\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)
$$

Some definitions:

- $\text{depth}(E) \leq D \implies$ finite # of trees
- $\text{length}(x_t) = D$
- $\|E\| : = |\{v \in \text{nodes}(E)\}|$
- $\|E\|_D : = \|E\| - |\{v \in \text{leaves}(E) : \text{depth}(v) = D\}|$
- $N|\text{leaves}(E)| = \# \text{ of tree experts } E \text{ for binary tree } T$
**Tree Experts: Setting**

Regret against tree expert $E$:

$$\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)$$

Some definitions:

- $\text{depth}(E) \leq D \implies$ finite # of trees
- $\text{length}(x_t) = D$
- $\|E\| := |\{v \in \text{nodes}(E)\}|$
- $\|E\|_D := \|E\| - |\{v \in \text{leaves}(E) : \text{depth}(v) = D\}|$
- $N|\text{leaves}(E)| = \# \text{ of tree experts } E \text{ for binary tree } T$
- $u \sqsubseteq v : \text{ length}(u) \leq \text{ length}(v), u_1 = v_1, \cdots, u_d = v_d$
Tree Experts: Setting

Regret against tree expert E:

$$\bar{R}_{E,T} = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell(i_E(x_t), y_t)$$

Some definitions:

- $\text{depth}(E) \leq D \implies$ finite # of trees
- $\text{length}(x_t) = D$
- $\|E\| := |\{v \in \text{nodes}(E)\}|$
- $\|E\|_D := \|E\| - |\{v \in \text{leaves}(E) : \text{depth}(v) = D\}|$
- $N_{\text{leaves}(E)} = \# \text{ of tree experts } E \text{ for binary tree } T$
- $u \sqsubseteq v : \text{ length}(u) \leq \text{ length}(v), u_1 = v_1, \ldots, u_d = v_d$
- $u \sqsubset v : \text{ length}(u) < \text{ length}(v), u_1 = v_1, \ldots, u_d = v_d$
TREE EXPERTS: HEDGE ALGORITHM

infeasible: \( M = N^{2D} \) tree experts \( \implies \) complexity \( \propto N^{2D} \)
TREE EXPERTS: HEDGE ALGORITHM

infeasible: $M = N^{2D}$ tree experts $\implies$ complexity $\propto N^{2D}$

In theory:
TREE EXPERTS: HEDGE ALGORITHM

infeasible: \( M = N^{2^D} \) tree experts \( \iff \) complexity \( \propto N^{2^D} \)

In theory:

- initialize: \( w_{E,0} = 2^{-\|E\|_D} N^{-|\text{leaves}(E)|} \)
Tree Experts: Hedge Algorithm

infeasible: \( M = N^{2D} \) tree experts \( \implies \) complexity \( \propto N^{2D} \)

In theory:

- **initialize**: \( w_{E,0} = 2^{-\|E\|_D} N^{-\text{leaves}(E)} \)
- **define**: \( w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1} \)
Tree Experts: Hedge Algorithm

infeasible: \( M = N^{2D} \) tree experts \( \implies \) complexity \( \propto N^{2D} \)

In theory:

- initialize: \( w_{E,0} = 2^{-\|E\|_D} N^{-|\text{leaves}(E)|} \)
- define: \( w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1} \)
- update weight of leaf \( v \) in \( E \):

\[
    w_{E,v,t} = \begin{cases} 
        w_{E,v,t-1} \exp(-\gamma \ell(i_E(v), y_t)) & \text{if } v \sqsubseteq x_t \\
        w_{E,v,t-1} & \text{otherwise}
    \end{cases}
\]
Tree Experts: Hedge Algorithm

infeasible: $M = N^{2D}$ tree experts $\implies$ complexity $\propto N^{2D}$

In theory:

- initialize: $w_{E,0} = 2^{-\|E\|_D} N^{-|\text{leaves}(E)|}$
- define: $w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1}$
- update weight of leaf $v$ in $E$:

$$w_{E,v,t} = \begin{cases} 
  w_{E,v,t-1} \exp(-\gamma \ell(i_E(v), y_t)) & \text{if } v \sqsubseteq x_t \\
  w_{E,v,t-1} & \text{otherwise}
\end{cases}$$

- $v$ unique $\Rightarrow$ one leaf updated: $w_{E,t} = w_{E,t-1} e^{-\gamma \ell(i_E(x_t), y_t)}$
Tree Experts: Hedge Algorithm

infeasible: $M = N^{2D}$ tree experts $\implies$ complexity $\propto N^{2D}$

In theory:

- initialize: $w_{E,0} = 2^{-\|E\|_D} N^{-|\text{leaves}(E)|}$
- define: $w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1}$
- update weight of leaf $v$ in $E$:

$$w_{E,v,t} = \begin{cases} w_{E,v,t-1} \exp(-\gamma \ell(i_E(v), y_t)) & \text{if } v \subseteq x_t \\ w_{E,v,t-1} & \text{otherwise} \end{cases}$$

- $v$ unique $\implies$ one leaf updated: $w_{E,t} = w_{E,t-1} e^{-\gamma \ell(i_E(x_t), y_t)}$
- Conditional distribution: $w_{k,t-1} = \sum_{E \mid i_E(x_t)=k} w_{E,t-1}$
Tree Experts: Hedge Algorithm

infeasible: $M = N^{2D}$ tree experts $\implies$ complexity $\propto N^{2D}$

In theory:

- initialize: $w_{E,0} = 2^{-\|E\|_D}N^{-|\text{leaves}(E)|}$
- define: $w_{E,t-1} = w_{E,0} \prod_{v \in \text{leaves}(E)} w_{E,v,t-1}$
- update weight of leaf $v$ in $E$:
  \[
  w_{E,v,t} = \begin{cases} 
    w_{E,v,t-1} \exp(-\gamma \ell(i_E(v), y_t)) & \text{if } v \subseteq x_t \\
    w_{E,v,t-1} & \text{otherwise}
  \end{cases}
  \]
- $v$ unique $\implies$ one leaf updated: $w_{E,t} = w_{E,t-1} e^{-\gamma \ell(i_E(x_t), y_t)}$
- Conditional distribution: $w_{k,t-1} = \sum_{E \mid i_E(x_t)=k} w_{E,t-1}$
- choose with probability: $p_{k,t} = \sum_{E \mid i_E(x_t)=k} w_{E,t-1} / \sum'_{E} w_{E',t-1}$
THE TREE EXPERT FORECASTER

**Parameters:** Real number $\eta > 0$, integer $D \geq 0$.

**Initialization:** $\overline{w}_{i,v,0} = 1$, $w_{i,v,0} = 1$ for each $i = 1, \ldots, N$ and for each node $v = (v_1, \ldots, v_d)$ with $d \leq D$.

For each round $t = 1, 2, \ldots$

1. draw an action $I_t$ from $\{1, \ldots, N\}$ according to the distribution
   $$p_{i,t} = \frac{\overline{w}_{i,\lambda,t-1}}{\sum_{j=1}^{N} \overline{w}_{j,\lambda,t-1}}, \quad i = 1, \ldots, N;$$

2. obtain $Y_t$ and compute, for each $v$ and for each $i = 1, \ldots, N$,
   $$w_{i,v,t} = \begin{cases} w_{i,v,t-1} e^{-\eta \ell(i,Y_t)} & \text{if } v \subseteq x_t \\ w_{i,v,t-1} & \text{otherwise} \end{cases}$$

3. recursively update each node $v = (v_1, \ldots, v_d)$ with $d = D, D-1, \ldots, 0$
   $$\overline{w}_{i,v,t} = \begin{cases} \frac{1}{2N} w_{i,v,t} & \text{if } v = x_t \\ \frac{1}{2N} \sum_{j=1}^{N} w_{j,v,t} & \text{if } \text{depth}(v) = D \\ \frac{1}{2N} w_{i,v,t} + \frac{1}{2N} (\overline{w}_{i,v_0,t} + \overline{w}_{i,v_1,t}) & \text{if } v \subsetneq x_t \\ \overline{w}_{i,v,t-1} & \text{if } \text{depth}(v) < D \\ \overline{w}_{i,v,t} & \text{if } v \not\subset x_t \end{cases}$$

where $v_0 = (v_1, \ldots, v_d, 0)$ and $v_1 = (v_1, \ldots, v_d, 1)$. 


Theorem 5.4. Regret bound of Hedge algorithm over tree experts of depth at most $D$:

\[
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \frac{2^D}{\gamma} \ln(2N) + \frac{\gamma T}{8}
\]

\[
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \sqrt{\frac{T}{2} 2^D \ln(2N)}
\]
Theorem 5.4. Regret bound of Hedge algorithm over tree experts of depth at most $D$:

\[
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \frac{2^D}{\gamma} \ln(2N) + \frac{\gamma T}{8}
\]

\[
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \sqrt{\frac{T}{2} 2^D \ln(2N)}
\]

- Recall: $M = N^{2^D} \iff \text{bound} = \sqrt{\frac{T}{2} \ln M} = \sqrt{\frac{T}{2} 2^D \ln N}$
Tree Experts

**Theorem 5.4.** Regret bound of Hedge algorithm over tree experts of depth at most $D$:

$$
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \frac{2^D}{\gamma} \ln(2N) + \frac{\gamma T}{8}
$$

$$
\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \sqrt{\frac{T}{2} 2^D \ln(2N)}
$$

- Recall: $M = N^{2^D}$ \implies bound $= \sqrt{\frac{T}{2} \ln M} = \sqrt{\frac{T}{2} 2^D \ln N}$
- Three expert algorithm has $N(2^{D+1} - 1)$ weights
**Tree experts**

**Theorem 5.4.** Regret bound of Hedge algorithm over tree experts of depth at most $D$:

$$\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \frac{2^D}{\gamma} \ln(2N) + \frac{\gamma T}{8}$$

$$\max_{E : \text{depth}(E) \leq D} \bar{R}_{E,T} \leq \sqrt{\frac{T}{2}} 2^D \ln(2N)$$

- Recall: $M = N^{2^D} \implies \text{bound} = \sqrt{\frac{T}{2}} \ln M = \sqrt{\frac{T}{2}} 2^D \ln N$

- Three expert algorithm has $N(2^{D+1} - 1)$ weights

**Theorem 5.5.** Distribution of the action $I_t$ by the tree expert forecaster = distribution of action $I'_t$ by the Hedge algorithm (with initialization $w_{E,0} = 2^{-\|E\|_D} N^{-|\text{leaves}(E)|}$).
SHORTEST PATH PROBLEM
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
- representation: $i \in \text{paths}(u, v) \subset \{0, 1\}^{|E|}$
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
- representation: $i \in \text{paths}(u, v) \subset \{0, 1\}^{|E|}$
- assume $\forall e \in E, \exists i \in \text{paths}(u, v) | \text{“} e \in i \text{”}$
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
- representation: $i \in \text{paths}(u, v) \subset \{0, 1\}^{|E|}$
- assume $\forall e \in E, \exists i \in \text{paths}(u, v) \mid \text{“e} \in i\text{”}$
- outcome $y_t = \text{vector of losses } \ell_t \in [0, 1]^{|E|}$ (j-th entry $= \ell_{e_j,t}$)
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
- representation: $i \in \text{paths}(u, v) \subset \{0, 1\}^{|E|}$
- assume $\forall e \in E, \exists i \in \text{paths}(u, v) \mid \text{"e } \in \text{ i"}$
- outcome $y_t = \text{vector of losses } \ell_t \in [0, 1]^{|E|}$ (j-th entry = $\ell_{e_j, t}$)
- loss of path $i$: $\ell(i, y_t) = i \cdot \ell_t$
SHORTEST PATH PROBLEM

- directed acyclic graph with vertices $V$, edges $E = \{e_i\}_i$
- paths from $u$ to $v$: $e^{(1)} = (u, v_1), \ldots, e^{(k)} = (v_{k-1}, v)$
- exponential number of path experts, but structured on a graph
- representation: $i \in \text{paths}(u, v) \subset \{0, 1\}^{|E|}$
- assume $\forall e \in E, \exists i \in \text{paths}(u, v) | "e \in i"$
- outcome $y_t = \text{vector of losses } \ell_t \in [0, 1]^{|E|}$ (j-th entry $= \ell_{e_j,t}$)
- loss of path $i$: $\ell(i, y_t) = i \cdot \ell_t$
- expected regret against shortest path:

$$\sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{i \in \text{paths}(u, v)} \sum_{t=1}^{T} \ell(i, y_t)$$

$$\sum_{t=1}^{T} \ell(\hat{p}_t, y_t) - \min_{i \in \text{paths}(u, v)} i \cdot \sum_{t=1}^{T} \ell_t$$
Follow the Perturbed Leader
FOLLOW THE PERTURBED LEADER

- $Z_1, \ldots, Z_T$ i.i.d. random vectors $\in \mathbb{R}^{|E|}$
**Follow the Perturbed Leader**

- $Z_1, \cdots, Z_T$ i.i.d. random vectors $\in \mathbb{R}^{|E|}$
- Forecaster chooses:

$$I_t = \arg\min_i \sum_{s=1}^{t-1} \ell_s + Z_t$$
Follow the Perturbed Leader

- $Z_1, \cdots, Z_T$ i.i.d. random vectors $\in \mathbb{R}^{|E|}$
- Forecaster chooses:

$$I_t = \arg\min_{i \in \text{paths}(u,v)} \left( \sum_{s=1}^{t-1} \ell_s + Z_t \right)$$

- efficient linear-time algorithms for shortest path problems
**Follow the Perturbed Leader**

- $Z_1, \cdots, Z_T$ i.i.d. random vectors $\in \mathbb{R}^{|E|}$
- Forecaster chooses:
  $$I_t = \arg\min_{i \in \text{paths}(u,v)} \left( \sum_{s=1}^{t-1} \ell_s + Z_t \right)$$

- efficient linear-time algorithms for shortest path problems
- apply them to forecaster
**Follow the Perturbed Leader**

- $Z_1, \ldots, Z_T$ i.i.d. random vectors $\in \mathbb{R}^{|E|}$
- Forecaster chooses:
  
  $$I_t = \arg\min_{i \in \text{paths}(u,v)} i \cdot \left( \sum_{s=1}^{t-1} \ell_s + Z_t \right)$$

- efficient linear-time algorithms for shortest path problems
- apply them to forecaster
- good bounds for the forecaster for $Z_t \sim U([0, \Delta]^{|E|})$
HEDGE ALGORITHM FOR SHORTEST PATH
HEDGE ALGORITHM FOR SHORTEST PATH

- In theory, choose among exponential # of path experts:

\[ p_{i,t} = \exp \left( -\gamma \sum_{s=1}^{t-1} i \cdot \ell_s \right) / \sum_{i' \in \text{paths}(u,v)} \exp \left( -\gamma \sum_{s=1}^{t-1} i' \cdot \ell_s \right) \]
HEDGE ALGORITHM FOR SHORTEST PATH

- In theory, choose among exponential # of path experts:

\[ p_{i,t} = \exp \left( -\gamma \sum_{s=1}^{t-1} i \cdot \ell_s \right) / \sum_{i' \in \text{paths}(u,v)} \exp \left( -\gamma \sum_{s=1}^{t-1} i' \cdot \ell_s \right) \]

- Cumulative loss of expert i:

\[ \sum_{s=1}^{t} i \cdot \ell_s = \sum_{s=1}^{t} \sum_{e \in i} \ell_{e,s} = \sum_{e \in i} L_{e,t} \]

with \( L_{e,t} = \sum_{s=1}^{t} \ell_{e,s} = \text{cumulative loss by edge } e \)
HEDGE ALGORITHM FOR SHORTEST PATH

- In theory, choose among exponential # of path experts:

\[ p_{i,t} = \exp \left( -\gamma \sum_{s=1}^{t-1} i \cdot \ell_s \right) / \sum_{i' \in \text{paths}(u,v)} \exp \left( -\gamma \sum_{s=1}^{t-1} i' \cdot \ell_s \right) \]

- Cumulative loss of expert i:

\[ \sum_{s=1}^{t} i \cdot \ell_s = \sum_{s=1}^{t} \sum_{e \in i} \ell_{e,s} = \sum_{e \in i} L_{e,t} \]

with \( L_{e,t} = \sum_{s=1}^{t} \ell_{e,s} = \text{cumulative loss by edge } e \)

- idea: Hedge algorithm on \(|V|\) edges and construct path by choosing edges one by one
HEDGE ALGORITHM FOR SHORTEST PATH
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \[ \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_{e,t} \]
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \( \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_{e,t} \)
- weight of vertex \( w \) at time \( t \):
  \[
  G_t(w) = \sum_{i \in \text{paths}(w,v)} \exp \left( -\gamma \sum_{e \in i} L_{e,t} \right)
  \]
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \( \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_{e,t} \)
- weight of vertex \( w \) at time \( t \):
  \[
  G_t(w) = \sum_{i \in \text{paths}(w,v)} \exp \left( -\gamma \sum_{e \in i} L_{e,t} \right)
  \]
- weights can be computed efficiently in \( O(|E|) \)
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \[ \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_e,t \]
- weight of vertex \( w \) at time \( t \):
  \[ G_t(w) = \sum_{i \in \text{paths}(w,v)} \exp \left( -\gamma \sum_{e \in i} L_e,t \right) \]
- weights can be computed efficiently in \( O(|E|) \)
- for \( i \in \text{paths}(u,v) \), let \( v_{i,k} = k\text{-th vertex along } i \)
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \( \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_{e,t} \)
- weight of vertex \( w \) at time \( t \):
  \[
  G_t(w) = \sum_{i \in \text{paths}(w,v)} \exp \left( -\gamma \sum_{e \in i} L_{e,t} \right)
  \]
- weights can be computed efficiently in \( O(|E|) \)
- for \( i \in \text{paths}(u,v) \), let \( v_{i,k} = k\text{-th vertex along } i \)
- \( I_t \) has distribution \( \Pi_k \mathbb{P}_t[v_{I_t,k} \mid v_{I_t,k-1}, \ldots, v_{I_t,0}] \)
HEDGE ALGORITHM FOR SHORTEST PATH

- recall: \( \sum_{s=1}^{t} i \cdot \ell_s = \sum_{e \in i} L_{e,t} \)
- weight of vertex \( w \) at time \( t \):
  \[
  G_t(w) = \sum_{i \in \text{paths}(w,v)} \exp \left( -\gamma \sum_{e \in i} L_{e,t} \right)
  \]
- weights can be computed efficiently in \( O(|E|) \)
- for \( i \in \text{paths}(u,v) \), let \( v_{i,k} = k\)-th vertex along \( i \)
- \( I_t \) has distribution \( \Pi_k P_t[v_{I_t,k} | v_{I_t,k-1}, \cdots, v_{I_t,0}] \)
- we have
  \[
  P_t[v_{I_t,k} | v_{I_t,k-1}, \cdots, v_{I_t,0}] = \begin{cases} 
    G_{t-1}(v_{I_t,k})/G_{t-1}(v_{I_t,k-1}) & \text{if } (v_{I_t,k-1}, v_{I_t,k}) \in E \\
    0 & \text{otherwise}
  \end{cases}
  \]