Building a good cache by combining the initial segments of two lists

\[ |T_1| + |T_2| = c \]

Arcing: Heuristic for setting a goal combination

- No refetching
  Adjust cache towards goal
- W. refetching
  Set current cache to goal

Relative loss bounds

\( M_{\text{ALG}}(S) \leq \alpha M^*(S) + \text{small additional terms} \)

Ideally 1
ONE EXPERT PER COMBINED CACHE

LOSES E [0, 13] HIT OR MISS

PROBLEM:
- HOW DO WE PRODUCE A CACHE FROM THE WEIGHT VECTOR OVER EXPERTS
- CAN'T USE WEIGHTED AVERAGE

\[ \bar{w}_t \cdot x_t \]

\[ \uparrow \text{BINARY, ONLY AVAILABLE AFTER REQUEST} \]
$n+1$ weights $w_0, \ldots, w_m$ on gaps

Idea 2: $y_t = \sum_{i=0}^{m} w_{t_i} \cdot i$  \( \text{mean} \)

- Why $i$?

- When miss, need to halve or total

weight $by: \beta = e^{-n}$

$w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ \ldots \ w_m$

$0 \ 2 \ 3 \ 4 \ \ldots \ m$

$\text{gaps}$

$\text{mean: } \alpha = \sum_{i=0}^{m} w_{t_i} \cdot i$

Left

Right

Total

Total

Not necessarily

50 50

Split
Can this be fixed?
Different potential?

Idea 2: Predict $W$, Weighted Median

Weight
Weights on gaps of left list

$w_0 \cdot w_1 \cdot w_2 \cdot \ldots \cdot w_{i-1} \cdot w_i \cdot w_{i+1} \cdot \ldots \cdot w_{n-1} \cdot w_n$

$\geq \frac{1}{2}$

Mean

Weights on gaps

$\Rightarrow$

Median

$\Rightarrow$

$n = 5$

Either Median upwards
Or "downwards multiplied by $1/2" \geq \frac{1}{2}$

Bounds work fine for AEP case

$M_n \leq 2M^* + O(\sqrt{\log\frac{1}{\alpha}}(\log n) + \log n)$

Ideally Median does not move too fast?
**Randomized Hedge Alg.**

- \( i \sim W_i \)
- Constant of \( i \) in front of \( M_i^k \)
- Too much re-fetching

**Open:** Is constant of \( i \) possible

\[ E(M_A) \leq 1 \cdot M_i^k + O\left( \sqrt{M_i^k \ln n} + \ln n \right) \]

\( k \) lists

\[ \sum_{i} \text{Initial Seg}_i = n \]

\( O(n^{\frac{k-1}{2}}) \) experts

**Hedge Alg. w. one weight per composite cache**
- Dynamic prog. \( O(kn^2) \)
- Fancy \( O(kn \log n) \)
- Too much re-fetching

For any determ. Alg \( \# \) misses can be \( \geq k \cdot M_i^k \)