Combining Initial Segments of Lists

Manfred K. Warmuth[1] Wouter M. Koolen[23] David P. Helmbold[1]

[1]University of California, Santa Cruz

[2]Royal Holloway, University of London

[3]Centrum Wiskunde en Informatica, Amsterdam

Algorithmic Learning Theory 2011

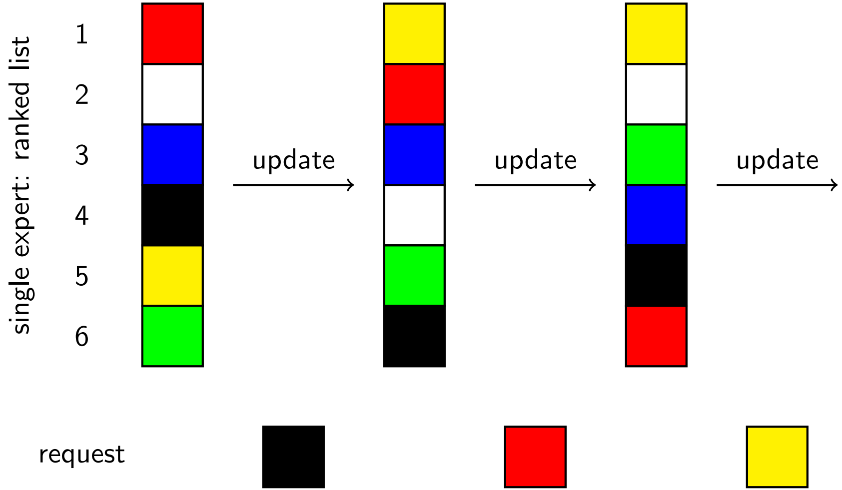
Problem

We want to help the user choose a color
We have access to intelligent palettes (our “experts”)

- gray levels
- pastels
- “Web colors”
- flags of the world
- copper tones
- ...

and we want a master algorithm to combine their advice.

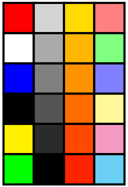
Intelligent palette example: flags of the world



Combining palettes

Not *one* but *several* intelligent palettes

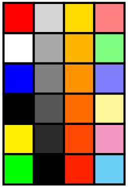
4 palettes



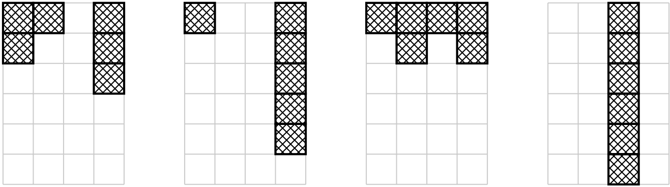
Combining palettes

Not *one* but *several* intelligent palettes

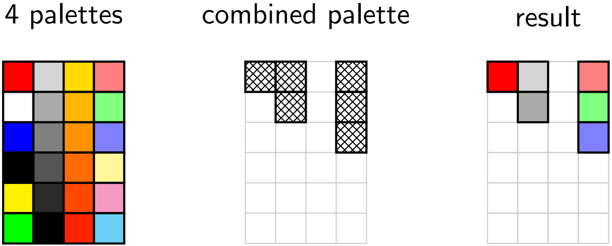
4 palettes



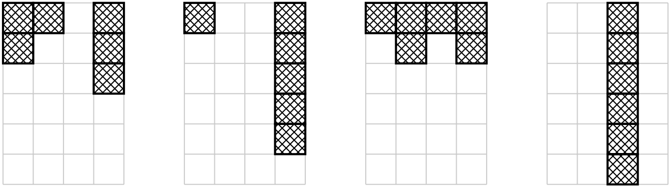
A combined palette consists of 6 slots from tops of “expert” palettes



Combining palettes

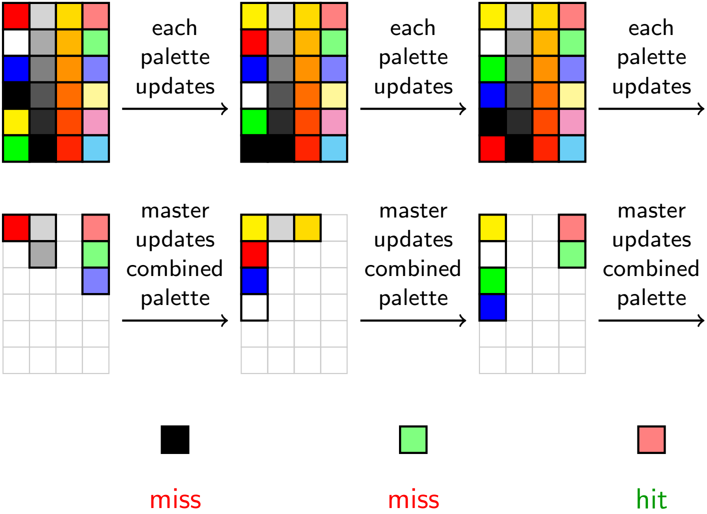


A combined palette consists of 6 slots from tops of “expert” palettes



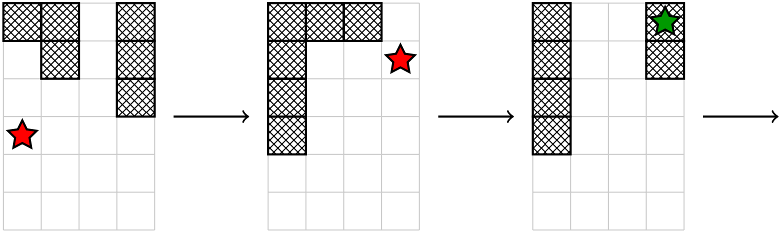
Master algorithm

A master algorithm chooses a combined palette each round.



A good master

A master algorithm chooses a combined palette each round.



Loss A combined palette incurs loss if it misses the requested item.

Goal Small regret compared to the best fixed combined palette.

$$(\text{regret} := \# \text{ misses of master} - \# \text{ misses of best combined palette})$$

Other Applications

- Meta-search, i.e. combining search engine results
- Meta-recommender
- Caching / paging

theory ⊢——⊣ ↑ BIG GAP ⊢——⊣ practice

this work

Abstract Problem

Setting We have K base lists of N slots each. Each round

- The base lists reveal their content.
- We select N items by taking initial segments of base lists.
- An item is requested. We either *hit* or *miss* it.

Goal Small regret w.r.t. the best fixed *combined list* in hindsight.

Difficulty There are $\binom{N+K-1}{K-1} \approx \left(\frac{N}{K}\right)^K$ such combined lists.

Abstract Problem

Setting We have K base lists of N slots each. Each round

- The base lists reveal their content.
- We select N items by taking initial segments of base lists.
- An item is requested. We either *hit* or *miss* it.

Goal Small regret w.r.t. the best fixed *combined list* in hindsight.

Difficulty There are $\binom{N+K-1}{K-1} \approx \left(\frac{N}{K}\right)^K$ such combined lists.

N	K	combined lists
10	20	10^7
100	20	$5{\cdot}10^{21}$

Summary of Results I

Offline problem Finding the best combined list in hindsight is

- Easy: $O(KN^2)$ without duplicate items.
- Hard: decision problem is *NP complete* with duplicates.
  (Is there a combined list with $\leq m$ misses for given items/requests?)

  Relaxed problem: replace miss *indicator* with miss *count*
  Not even close. (That's why it works.)

Online implementation Simulate (randomized) Hedge algorithm in $O(K\,N \log N)$ time per trial using Fast Fourier Transform.
Faster than offline!

Summary of Results II

Regret The expected regret of Hedge is at most

$$\sqrt{BK \ln \frac{N}{K}},$$

where B is the loss of the best combined list.

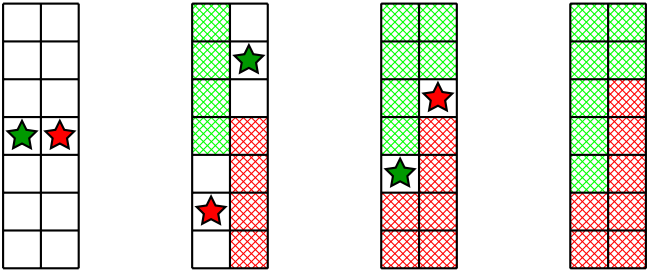
Lower bound Any algorithm has regret at least

$$\sqrt{\max\{\underbrace{B \ln N}_{K \text{ missing}}, \underbrace{BK}_{\ln \frac{N}{K} \text{ missing}}\}}.$$

Open problem Gap between bounds

Lower bound flavour

$K = 2$ base lists. **Reduction** to series of 2-expert games.
Key every combined list must miss one of the star items.



- $S = \log_2(N+1)$ many phases.
- each phase, enforce loss $B/S + \sqrt{\frac{B}{S\pi}}$.
- master loss is $B + \sqrt{B \log_2(N+1)/\pi}$.
- best combined list has loss B.

Conclusion

- Combining initial segments of lists
- Without duplicates:
  - Efficient implementation of Hedge algorithm
  - Faster than offline
  - Fast Fourier Transform
- With duplicates:
  - Hardness result
  - Transition to miss counts (cf attribute errors)
- Regret bound
- Two complementary* lower bounds