Combining Initial Segments of Lists

Manfred K. Warmuth **Wouter M. Koolen** David P. Helmbold

UNIVERSITY OF CALIFORNIA SANTA CRUZ
Royal Holloway University of London
UNIVERSITY OF CALIFORNIA SANTA CRUZ

University College London
16-2-12

Extended abstract in ALT 2011 – Journal version in preparation
Supported by NWO Rubicon grant 680-50-1010

Outline

Motivating Applications

Many methods/heuristics produce ranked lists

- Search engines
- Item recommenders
- Caching / paging schemes
- Text input completion
- ...

Luxury: we often have *several* rankers.

- Simple goal: learn which ranker is the best
- Our goal: learn to *combine* ranked lists

theory ⊢——⊣ ↑ BIG GAP ⊢——⊣ practice

this work

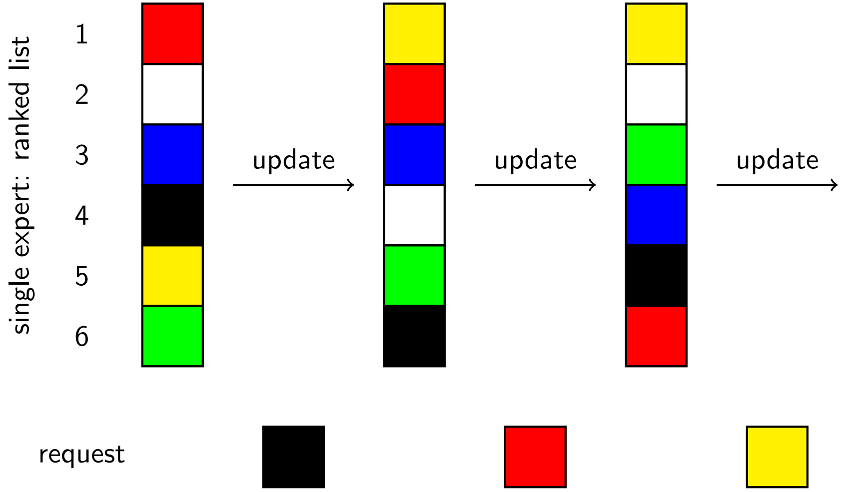
Example Problem

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.

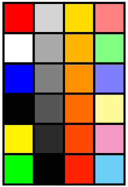
Intelligent palette example: flags of the world



Combining palettes

Not *one* but *several* intelligent palettes

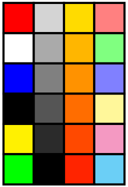
4 palettes



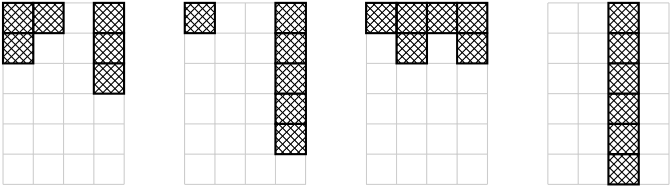
Combining palettes

Not *one* but *several* intelligent palettes

4 palettes

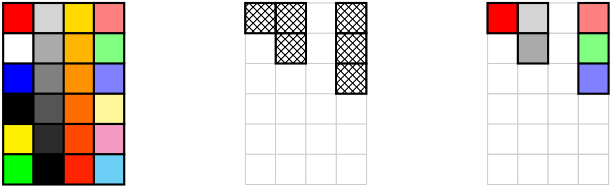


A combined palette consists of 6 slots from tops of "expert" palettes

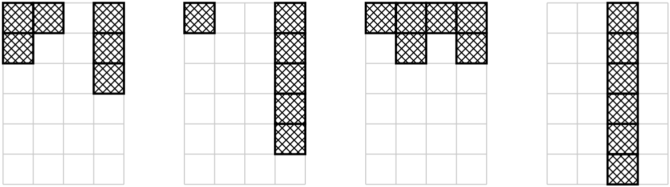


Combining palettes

4 palettes | combined palette | result



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



Abstract Problem

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

- 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.

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

Goal Small regret compared to the best fixed combined list in hindsight

$$\left(\text{regret} \;:=\; \#\text{ misses of master} - \#\text{ misses of best combined list}\right)$$

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 trial

- 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.

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

Goal Small regret compared to the best fixed combined list in hindsight

$$\left(\text{regret} \;:=\; \#\text{ misses of master} - \#\text{ misses of best combined list}\right)$$

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

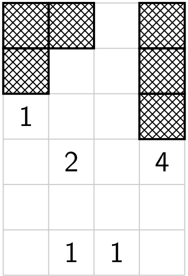
N	K	combined lists
10	20	10^7
100	20	$5 \cdot 10^{21}$

Outline

What's so special about absence of duplicates?

Loss of a combined list is the *independent sum* of K base-list-wise losses.

Cumulative slot request counts:



Loss:

$$1 + 3 + 1 + 4$$

The Hedge Algorithm (Freund & Schapire, 1997)

Definition

Fix learning rate $\beta \in (0,1]$. Let combined lists $c_1, c_2, \ldots$ have cumulative losses $M_1, M_2, \ldots$ Hedge plays combined list c_i with probability $\frac{\beta^{M_i}}{\sum_j \beta^{M_j}}$.

The Hedge Algorithm (Freund & Schapire, 1997)

Definition

Fix learning rate $\beta \in (0, 1]$. Let combined lists $c_1, c_2, \ldots$ have cumulative losses $M_1, M_2, \ldots$ Hedge plays combined list c_i with probability $\frac{\beta^{M_i}}{\sum_j \beta^{M_j}}$.

Various schemes exist for tuning the parameter β.

The Hedge Algorithm (Freund & Schapire, 1997)

Definition

Fix learning rate $\beta \in (0, 1]$. Let combined lists $c_1, c_2, \ldots$ have cumulative losses $M_1, M_2, \ldots$ Hedge plays combined list c_i with probability $\frac{\beta^{M_i}}{\sum_j \beta^{M_j}}$.

Various schemes exist for tuning the parameter β.

Too many combined lists to follow this recipe explicitly.

Teaser: Offline problem

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $B_{n,k}$ to be the loss of the best partial combined list taking n elements from the tops of the first k list. We then have the recurrence

$$B_{n,k} = \min_{0 \leq c \leq n} (M_{c,k} + B_{n-c,k-1}) \qquad B_{n,0} = \begin{cases} 0 & n = 0 \\ \infty & n > 0 \end{cases}$$

The offline loss $B_{N,K}$ can be computed in time $O(N^2 K)$.

Teaser: Offline problem

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $B_{n,k}$ to be the loss of the best partial combined list taking n elements from the tops of the first k list. We then have the recurrence

$$B_{n,k} = \min_{0 \leq c \leq n} (M_{c,k} + B_{n-c,k-1}) \qquad B_{n,0} = \begin{cases} 0 & n = 0 \\ \infty & n > 0 \end{cases}$$

The offline loss $B_{N,K}$ can be computed in time $O(N^2 K)$.
$B_{\star,k}$ is *infimal convolution* of $M_{\star,k}$ and $B_{\star,k-1}$: speedup to $O(\frac{N^2}{\ln N} K)$.

Teaser: Offline problem

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $B_{n,k}$ to be the loss of the best partial combined list taking n elements from the tops of the first k list. We then have the recurrence

$$B_{n,k} = \min_{0 \le c \le n} (M_{c,k} + B_{n-c,k-1}) \qquad B_{n,0} = \begin{cases} 0 & n = 0 \\ \infty & n > 0 \end{cases}$$

The offline loss $B_{N,K}$ can be computed in time $O(N^2 K)$.
$B_{\star,k}$ is *infimal convolution* of $M_{\star,k}$ and $B_{\star,k-1}$: speedup to $O(\frac{N^2}{\ln N} K)$.

Exponentially many combined lists. Why can we do this?
Loss of a combined list is the *independent sum* of K base-list-wise losses.

Implicit Hedge

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $Z_{n,k}$ to be the sum of β^{loss} of all partial combined lists taking n elements from the tops of the first k list. We then have the recurrence

$$Z_{n,k} = \sum_{0 \leq c \leq n} \beta^{M_{c,k}} Z_{n-c,k-1} \qquad Z_{n,0} = \begin{cases} 1 & n = 0 \\ 0 & n > 0 \end{cases}$$

The Hedge normalisation $Z_{N,K}$ can be computed in time $O(N^2 K)$.

Implicit Hedge

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $Z_{n,k}$ to be the sum of β^{loss} of all partial combined lists taking n elements from the tops of the first k list. We then have the recurrence

$$Z_{n,k} = \sum_{0 \le c \le n} \beta^{M_{c,k}} Z_{n-c,k-1} \qquad Z_{n,0} = \begin{cases} 1 & n = 0 \\ 0 & n > 0 \end{cases}$$

The Hedge normalisation $Z_{N,K}$ can be computed in time $O(N^2 K)$.
$Z_{\star,k}$ is *convolution* of $\beta^{M_{\star,k}}$ and $Z_{\star,k-1}$: speedup to $O(N \ln(N) K)$.

Implicit Hedge

K base lists of N slots each.
$M_{c,k}$ number of requests in base list k at position $> c$

Define $Z_{n,k}$ to be the sum of β^{loss} of all partial combined lists taking n elements from the tops of the first k list. We then have the recurrence

$$Z_{n,k} = \sum_{0 \le c \le n} \beta^{M_{c,k}} Z_{n-c,k-1} \qquad Z_{n,0} = \begin{cases} 1 & n = 0 \\ 0 & n > 0 \end{cases}$$

The Hedge normalisation $Z_{N,K}$ can be computed in time $O(N^2 K)$.
$Z_{\star,k}$ is *convolution* of $\beta^{M_{\star,k}}$ and $Z_{\star,k-1}$: speedup to $O(N \ln(N) K)$.

Given $Z_{\star,\star}$ table we can sample from the Hedge distribution in $O(NK)$.

Regret Bounds

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

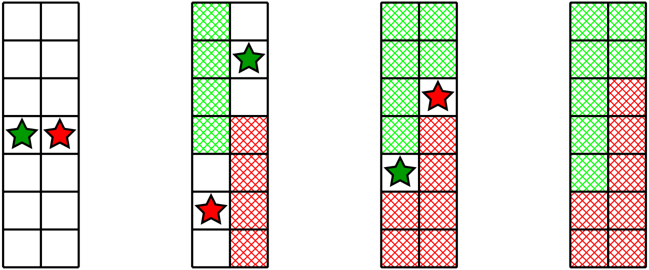
$$\sqrt{\max\{\underbrace{B \ln N}_{K \text{ missing}}, \underbrace{BK}_{\ln \frac{N}{K} \text{ missing}}\}}.$$

Open problem Gap between bounds

Deterministic algorithms have regret $\Omega(B(K-1))$

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.

Outline

Bleak

Duplicate items: computational complexity exhibits phase transition.

Decision problem Given a sequence of base list contents and requests. Is there a combined list with no misses?

is *NP complete*.

No hope for efficient algorithms with low regret unless RP = NP.

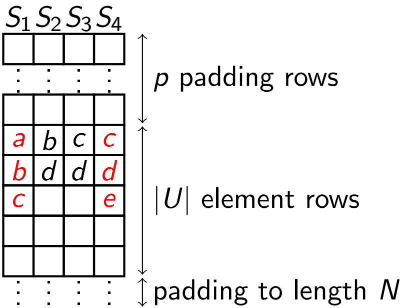
Relaxed problem replace miss *indicator* with miss *count*
Not even close. (That's why it works.)
Our algorithms work for this loss.

Reduction from set cover problem

Deciding whether a combined list with no loss exists is NP hard.

$$U = \{a, b, c, d, e\}$$
$$\mathcal{C} = \{S_1, S_2, S_3, S_4\}$$
$$S_1 = \{a, b, c\}$$
$$S_2 = \{b, d\}$$
$$S_3 = \{c, d\}$$
$$S_4 = \{c, d, e\}$$
$$m = 2$$

$\Rightarrow$



$K = |\mathcal{C}|$ base lists of size $N = m(p + |U|)$ with padding $p = m(|U| - 1)$, and a sequence of $|U|$ requests.

Circumventing the NP-completeness

Learning when lists have duplicates

- Same algorithm motivated by dot loss a factor of K
- Regret has additional factor of K because of overcount of misses

$$\sqrt{\underbrace{\overbrace{B}^{\text{misses}}\ K}_{\text{dot loss}}\ \ \underbrace{K\log n}_{\#\text{ of comb.lists}}}$$

Parallels regret bound for Winnow algorithm for learning disjunctions

- Switch from classification to attribute error introduces factor of K

$$\sqrt{\underbrace{\overbrace{B}^{\text{class.error}}\ \mathbf{K}}_{\text{attr.error}}\ \ \underbrace{K\log n}_{\#\text{ of disj.}}}$$

Outline

Conclusion

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