the hedge algorithm
and applications

Virgil Pavlu
will it rain tomorrow?

- I say it will rain
- Nick says it will not rain
- CNN says it will rain
- FOX says it will not rain

- TOMORROW: sunny all day…
- what about Saturday?
weighted majority algorithm

- Give the “weather-men” weights
  - Initial uniform or according to some prior belief

- Run the forecast for several days. Every day:
  - Make our prediction by weighted-majority-vote
  - Get the real outcome
  - Update weights
    - “penalize” wrong predictors
    - “reward” good predictors
more problems

- more problems
  - trading stocks
  - IR metasearch
  - disease classification
    - where we need to “train” doctors

- common underlying idea
  - generalization of weighted-majority

- why *not* exactly WM
  - losses are real numbers instead of discrete 0-1
  - our loss may be a weighted sum of expert losses
Hedge algorithm for online allocation [schapire, freund ’96]

Applications
- On classification: Adaboost
- On IR: metasearch algorithm
“how to use expert advice”?

- N experts (or strategies)
- maintain a set of weights over experts
- loop for T episodes $t=1,2,\ldots,T$
  - allocate Resources (believe) - based on weights
  - receive loses
  - reweight the experts
online allocation - hedge algorithm

\[ p_i^t = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t} \]
\[ l^t \in [0,1]^N \]
\[ L_H^t = p^t \cdot l^t \]
\[ w_i^{t+1} = w_i^t \cdot \beta \cdot l_i^t \]
\[ L_{\text{HEDGE}} = \sum_{t=1}^{T} p^t \cdot l^t \]

\( \beta \in [0,1] \); \( N \) strategies (systems)
initial weights \( w^1 \in [0,1]^N \); \( \sum_{i=1}^{N} w_i^1 = 1 \)
why hedge

- **goal**: Hedge loss close to the Loss of the best expert (bound)

- **proof idea**:
  - relate episodic Hedge loss with the sum of weights
    \[
    \sum_{i=1}^{N} w_{i}^{t+1} \leq \left( \sum_{i=1}^{t} w_{i}^{t} \right) \left( 1 - (1 - \beta)L_{Hedge}^{t} (\beta) \right)
    \]
  - relate cumulative Hedge loss with sum of final weights
    \[
    \ln \left( \sum_{i=1}^{N} w_{i}^{T+1} \right) \leq -(1 - \beta)L_{Hedge}(\beta)
    \]
  - relate sum of final weights with loss of the best expert
    \[
    L_{Hedge}(\beta) \leq \frac{\ln(1/\beta)}{1 - \beta} L_{best} + \frac{1}{1 - \beta} \ln(N)
    \]
\[
\sum_{i=1}^{N} w_i^{t+1} \leq \left( \sum_{i=1}^{N} w_i^t \right) \left( 1 - (1 - \beta) l_i^t \right)
\]

\[
\sum_{i=1}^{N} w_i^t = \sum_{i=1}^{N} w_i^t \beta^t l_i^t
\]

\[
\leq \sum_{i=1}^{N} w_i^t \left( 1 - (1 - \beta) l_i^t \right)
\]

\[
= \sum_{i=1}^{N} p_i^t \left( \sum_{i=1}^{N} w_i^t \right) \left( 1 - (1 - \beta) l_i^t \right)
\]

\[
= \left( \sum_{i=1}^{N} w_i^t \right) \sum_{i=1}^{N} \left( p_i^t - (1 - \beta) p_i^t l_i^t \right)
\]

\[
= \left( \sum_{i=1}^{N} w_i^t \right) \left( 1 - (1 - \beta) \sum_{i=1}^{N} p_i^t l_i^t \right)
\]

\[
= \left( \sum_{i=1}^{N} w_i^t \right) \left( 1 - (1 - \beta) p^t \cdot l^t \right)
\]

\[
\beta, l \in [0,1]
\]

\[
\beta^t \leq 1 - (1 - \beta) l
\]

\[
f(x) = 1 - (1 - \beta) x
\]

\[
f(x) = \beta^x
\]

\[
p_i^t = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t}
\]

\[
L^t_{Hedge} = p^t \cdot l^t
\]
\[
\ln \left( \sum_{i=1}^{N} w_i^{T+1} \right) \leq -(1 - \beta) L_{Hedge}(\beta)
\]

\[
\sum_{i=1}^{N} w_i^{t+1} \leq \left( \sum w_i^t \right) \left( 1 - (1 - \beta) L_t^{Hedge} \right)
\]

Applying repeatedly for \( t = 1, \ldots, T \) yields

\[
\sum_{i=1}^{N} w_i^{t+1} \leq \left( \sum w_i^1 \right) \prod_{t=1}^{T} \left( 1 - (1 - \beta) p^t \cdot l^t \right)
\]

\[
\leq \left( \sum w_i^1 \right) \prod_{t=1}^{T} \exp \left( - (1 - \beta) p^t \cdot l^t \right)
\]

\[
\leq \exp \left( -(1 - \beta) \sum_{t=1}^{T} p^t \cdot \ell^t \right)
\]

\[
L_{Hedge} = \sum_{t=1}^{T} p^t \cdot l^t
\]
[almost] as good as the best expert

\[
\ln \left( \sum_{i=1}^{N} w_i^{T+1} \right) \leq -(1 - \beta) L_{\text{Hedge}(\beta)} \\
\sum_{i=1}^{N} w_i^{T+1} \geq w_{\text{any}}^{T+1} = w_{\text{any}}^{1} \beta^{L_{\text{any}}} \\
w_{\text{any}}^{1} = \frac{1}{N}
\]

\[
L_{\text{Hedge}(\beta)} \leq \frac{-\ln \left( \frac{1}{N} \right) - L_{\text{any}} \ln \beta}{1 - \beta}
\]
Theorem 3 Let $B$ be an algorithm for the on-line allocation problem with an arbitrary number of strategies. Suppose that there exist positive real numbers $a$ and $c$ such that for any number of strategies $N$ and for any sequence of loss vectors $\ell^1, \ldots, \ell^T$

\[ L_B \leq cL_{\text{best}} + a\ln(N) \]

Then for all $\beta \in (0, 1)$, either

\[ c \geq \frac{\ln(1/\beta)}{1 - \beta} \quad \text{or} \quad a \geq \frac{1}{1 - \beta} \]
how to choose $\beta$

$$L_{Hedge}(\beta) \leq \frac{\ln(1/\beta)}{1-\beta} L_{best} + \frac{1}{1-\beta} \ln(N)$$

- $\beta = \text{confidence parameter}$
- think of it as a trade-off
- try to make the bound tight
- binary search : perfect expert + ($\beta=0$)
that wasn’t so bad

☑ Hedge algorithm for online allocation

- Applications
  - On classification: Adaboost
  - On IR: metasearch algorithm
• **disease classification**….
  • get past data
  • “train” a disease-predictor “doctor” (“hypothesis”, ”weak learner”)

• **how good is it** (on training data) ?
  • where is it wrong ?
  • train a new predictor to correct mistakes for the first one
hedge application: AdaBoost

**HEDGE**
- given: experts
- incoming: loses
- reweight: experts

**BOOSTING**
- given: datapoints
- incoming: weak learners
- reweight: datapoints

**ADABOOST**
\[
D_1(i) = \frac{1}{m}
\]

\[
h_t : X \to \{-1,1\}
\]

\[
\varepsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]
\]

\[
\beta_t = \frac{1}{\sqrt{\varepsilon_t}} > 1
\]

\[
D_{t+1} = \frac{D_t Z_t}{\sum_{i} \beta_t} \begin{cases} 
1/\beta_t & \text{if } y_i = h_t(x_i) \\
\beta_t & \text{if } y_i \neq h_t(x_i)
\end{cases}
\]

\[
H_{\text{final}}(x) = \text{sgn} \left( \sum_t \ln(\beta_t) h_t(x) \right)
\]
AdaBoost - example

- Start with uniform distribution on data
- Weak learners = halfplanes
round 1
round 2

$h_2$

$D_3$

$\varepsilon_2 = 0.21$

$\alpha_2 = 0.65$
$H_{\text{final}} = \text{sign}(0.42 + 0.65 + 0.92)$
AdaBoost - analysis

- **training error**

\[
\text{training error}(H_{\text{final}}) \leq \prod_t \sqrt{4\varepsilon_t (1 - \varepsilon_t)}
\]

- **generalization error**

\[
P_D [y f(x) \leq 0] \leq P_S [y f(x) \leq \theta] + O \left( \frac{1}{\sqrt{m}} \left( \frac{\log m \log |\mathcal{H}|}{\theta^2} + \log(1/\delta) \right)^{1/2} \right)
\]

\[
f(x) = \sqrt{4x(1-x)}, \ x \in [0,1]
\]
The prize was awarded to Yoav Freund and Robert Schapire for their paper "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences 55 (1997), pp. 119-139. This paper introduced AdaBoost, an adaptive algorithm to improve the accuracy of hypotheses in machine learning. The algorithm demonstrated novel possibilities in analysing data and is a permanent contribution to science even beyond computer science. Because of a combination of features, including its elegance, the simplicity of its implementation, its wide applicability, and its striking success in reducing errors in benchmark applications even while its theoretical assumptions are not known to hold, the algorithm set off an explosion of research in the fields of statistics, artificial intelligence, experimental machine learning, and data mining. The algorithm is now widely used in practice. The paper highlights the fact that theoretical computer science continues to be a fount of powerful and entirely novel ideas with significant and direct impact even in areas, such as data analysis, that have been studied extensively by other communities.”
what we did last year [Aslam, Pavlu, Savell]

- Hedge algorithm for online allocation

- Applications
  - On classification : Adaboost
  - On IR : metasearch algorithm
Search and metasearch

- **Search engines:**
  - Provide a ranked list of documents.
  - May provide relevance scores.

- **Metasearch engines:**
  - Query multiple search engines.
  - May or may not combine results.
metasearch: Dogpile

Search engine: Looksmart found 117 results.
The query string sent was: -chili -peppers

1. The Red Hot Chili Peppers
Find photos, lyrics, updates, tour info, and news on alternative-funk-rock band the Red Hot Chili Peppers.
Looksmart category – Red Hot Chili Pepper

2. Red Hot Chili Peppers Audio and Video
Watch videos and listen to music by this rock/funk band.
Looksmart category – Red Hot Chili Peppers

3. Chili and Hot Sauces
Shop for mouth-burning chili sauces, Tabasco, hot salsas and other pepper-inspired sauces.
Looksmart category – Chili & Hot Sauces

4. Chili and Hot Sauces
Find chili and other hot sauce recipes, including salsas, dips, spices, and rubs, and visit the Pepper Fool.
Looksmart category – Chili & Hot Sauces

5. Red Hot Chili Peppers – Screens and Themes
Promotional screensaver for the funk-rock band features falling chili peppers.
LookSmart category – Red Hot Chili Peppers Multimedia

Next set of results from Looksmart

Search engine: GoTo.com found 10 or more results.
metasearch: Metacrawler
metasearch: Profusion
metasearch algorithms

- **Heuristics and hacks:**
  - Interleave, average rank, sum scores, etc.

- **Principled models:**
  - Bayesian inference, election theory, etc.
  - *On-line combination of expert advice.*
hedge application: metasearch

HEDGE
- given: experts
- incoming: loses
- reweight: experts

METASEARCH
- given: search engines
- incoming: documents
  - compute losses
- reweight: search engines

Core: Hedge

SYSTEM WEIGHTS
MODIFY WEIGHTS

FEEDBACK
a unified model

Core : Hedge

Feedback

System weights

Modify weights
Core : Hedge

DOCUMENT SELECTOR

JUDGE

SYSTEM WEIGHTS

MODIFY WEIGHTS

a unified model
a unified model
ranks importance

- map ranks to values

\[
value(r) = \frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{Z} \approx \ln \frac{Z}{r}
\]

RELEVANT = -1
NONRELEVANT = +1

\[
LOSS(d, s) = label(d) \cdot value(rank_{d,s}) \approx label(d) \cdot \ln \frac{Z}{r}
\]

total loss vs. total precision vs. average precision
- **average_value at episode t**
  - “trust” in systems change with t
- **metasearch list**: order docs by
  - get feedback
  - compute losses
  - modify weights

**Feedback Loop**

\[
\text{average_value}_t(d) = \sum_{s=1}^{N} w^t_{s} \cdot \text{value}(\text{rank}_d,s)
\]

**Metasearch**

- **CIKM 2003 Homepage**
- **CIKM 2003 Call For Papers**
- **CIKM Home Page**
- **ACM DL: CIKM**
- **The Conference on Information and Knowledge Management**
- **ACM CIKM 2003 PRELIMINARY**
- **ACM CIKM Home Page**
- **ACM DL: CIKM**
- **ACM CIKM 2003 PRELIMINARY [Asis–I]**
- **CIKM 2003 (DBWORLD)**
- **CIKM 2003 (Pamphlet)**
- **bridge–cikm–2003**
- **Collaborative Filtering Mailing List**
- **Yahoo! Groups: webir**
- **dbForums - Cfp: Cikm 03**
- **Mailing List ARL–ERESERVE@arla.mpi**
- **Received: from cni.org by b.c.**
- **ACM WIDM 2003**
- **ACM – MMDB 2003**
- **Selected Publications**

**Core: Hedge**

- **DOCUMENT SELECTOR**
- **SYSTEM WEIGHTS**
- **JUDGE**
- **MODIFY WEIGHTS**
- **RANK-VALUE MAP**
experiments

- TREC 3,5,6,7,8
  - 41-129 systems
  - 50 queries per TREC
  - Metasearch combines all systems
- Use TREC judgments as user feedback
**metasearch - no feedback (yet)**

<table>
<thead>
<tr>
<th>TREC</th>
<th>MNZ</th>
<th>COND</th>
<th>Hedge-0</th>
<th>%MNZ</th>
<th>%COND</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.423</td>
<td>0.403</td>
<td>0.418</td>
<td>−1.2</td>
<td>+3.7</td>
</tr>
<tr>
<td>5</td>
<td>0.294</td>
<td>0.307</td>
<td>0.309</td>
<td>+5.1</td>
<td>+0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.341</td>
<td>0.315</td>
<td>0.345</td>
<td>+1.2</td>
<td>+9.5</td>
</tr>
<tr>
<td>7</td>
<td>0.320</td>
<td>0.308</td>
<td>0.323</td>
<td>+0.9</td>
<td>+4.9</td>
</tr>
<tr>
<td>8</td>
<td>0.350</td>
<td>0.343</td>
<td>0.352</td>
<td>+1.4</td>
<td>+2.6</td>
</tr>
</tbody>
</table>

MNZ = CombMNZ(Fox, Shaw, Lee et al)
COND = Condorcet(Aslam, Montague)
metasearch – TREC8

The diagram shows the metasearch performance for Hedge and TREC8 as a function of the number of documents judged.

- **Hedge**: Represented by a blue line. It starts with a steep increase, saturating at around 0.55 MAP as the number of documents judged increases.
- **TREC8**: Represented by a green dashed line, consistently at 0.45 MAP.

The x-axis represents the number of documents judged, ranging from 0 to 500. The y-axis measures MAP, ranging from 0.35 to 0.55.

The blue line indicates the performance of Hedge, which shows a significant improvement with the number of documents judged, approaching a plateau at a MAP of approximately 0.55.

The green dashed line represents TREC8, which maintains a constant MAP of 0.45 throughout the range of documents judged.

The red line labeled 'CombMNZ' is not visible in the description provided.
metasearch – TREC 3
metasearch – TREC 5

![Graph showing metasearch performance with Hedge and TREC5 results. The x-axis represents the number of docs judged, ranging from 0 to 500. The y-axis represents MAP, ranging from 0.3 to 0.4. The graph includes lines for Hedge, CombMNZ, and the best system.](image-url)
metasearch – TREC 6

Metasearch Performance
Hedge
TREC6

MAP

# docs judged

Hedge
CombMNZ
best system
Metasearch Performance – TREC 7

MAP vs # docs judged graph showing the performance of different methods.

- **Metasearch Performance**
- **Hedge**
- **TREC7**
metasearch – TREC 3, 5, 6, 7
actually...we do more than metasearch
actually...we do more than metasearch

system evaluation

"some" method evaluation

real evaluation
“a unified model for metasearch, pooling and system evaluation”
conclusion

- theoretic explanation of “being adaptive”
- simple, elegant, intuitive
- usually performs much better than the bound
END
AdaBoost - technical

- Start with uniform distrib $D_1$:
- At every round $t=1$ to $T$ given $D_t$
  - find weak hypothesis with error
  - compute "belief" in $h_t$
  - update distribution
- final hypothesis:

$$H_{\text{final}}(x) = \text{sgn} \left( \sum_t \alpha_t h_t(x) \right)$$

$$D_1(i) = \frac{1}{m}$$

$$h_t : X \rightarrow \{-1, 1\}$$

$$\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$$

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right) > 0$$

$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$
On-line Setup

Start with uniform distribution D

Choose Allocation

Receive loss vector → Update distribution

Hedge

FEEDBACK

SYSTEM WEIGHTS

MODIFY WEIGHTS
On-line Metasearch [Aslam, Pavlu, Savell]
hedge application : metasearch

- problem : metasearch
  - search engines
  - document judgment
  - metaserach

- [hedge] solution
  - search engines are “experts”
  - judgments on documents are “loses”
an IR setup
generalization error- based on margins

- training error [Schapire,Freund 1996]

$$\text{training error}(H_{\text{final}}) \leq \prod_t \sqrt{4\varepsilon_t (1 - \varepsilon_t)}$$

- generalization error [Schapire,Freund,Barlett,Lee 1998]

**Theorem 1** Let $\mathcal{D}$ be a distribution over $X \times \{-1, 1\}$, and let $S$ be a sample of $m$ examples chosen independently at random according to $\mathcal{D}$. Assume that the base-classifier space $\mathcal{H}$ is finite, and let $\delta > 0$. Then with probability at least $1 - \delta$ over the random choice of the training set $S$, every weighted average function $f \in \mathcal{C}$ satisfies the following bound for all $\theta > 0$:

$$P_{\mathcal{D}}[yf(x) \leq 0] \leq P_S[yf(x) \leq \theta] + O \left( \frac{1}{\sqrt{m}} \left( \frac{\log m \log |\mathcal{H}|}{\theta^2} + \log(1/\delta) \right)^{1/2} \right).$$
hedge approach
• map ranks to values

\[ \text{value}(r) = \frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{Z} \approx \ln \frac{Z}{r} \]

**RELEVANT** = -1
**NONRELEVANT** = +1

\[ \text{LOSS}(d, s) = \text{label}(d) \cdot \text{value}(\text{rank}_{d,s}) \approx \text{label}(d) \cdot \ln \frac{Z}{r} \]

\[ \text{TP}(s) = \text{total precision of } s = \text{average of precision values at all ranks} \]

FACT: \[ \sum_{\text{all docs}} \text{LOSS} \ (d, s) = C \times (Z - 2 \times \text{TP}(s)) \]

• Average the precision at **ALL** ranks
• Normalize so ideal system gets TP=1
• math is simpler
**Hedge**

- **average_value at episode t**
  - “trust” in systems change with t

- **metasearch** : include already judged docs

- **pooling**
  \[ d_t = \arg \max_d \left( \text{average}_\text{value}(d) \right) \]
  - system evaluation

- **get feedback**

- **modify weights**
  \[ w_{s}^{t+1} = w_{s}^{t} \cdot \beta^{\text{LOSS}_t (d_t, s)} \]

**average_value_t(d) =**
\[ = \sum_{s=1}^{N} w_{s}^{t-1} \cdot \text{value(rank}_{d,s}) \]
actually...we do more than metasearch

avg precision

feedback

online metasearch

best underlying system

metasearch
experiments

- TREC 3,5,6,7,8
  - 41-129 systems
  - 50 queries per TREC
  - metasearch uses all systems

- use TREC judgments as user feedback

- system evaluation: incomplete judgments
experiments - relevant docs found

Hedge

Cormack et al.

TREC depth pooling
experiments - system evaluation
system evaluation – kendall’s $\tau$
metasearch - no feedback (yet)

<table>
<thead>
<tr>
<th>TREC</th>
<th>MNZ</th>
<th>COND</th>
<th>Hedge-0</th>
<th>%MNZ</th>
<th>%COND</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.423</td>
<td>0.403</td>
<td>0.418</td>
<td>-1.2</td>
<td>+3.7</td>
</tr>
<tr>
<td>5</td>
<td>0.294</td>
<td>0.307</td>
<td>0.309</td>
<td>+5.1</td>
<td>+0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.341</td>
<td>0.315</td>
<td>0.345</td>
<td>+1.2</td>
<td>+9.5</td>
</tr>
<tr>
<td>7</td>
<td>0.320</td>
<td>0.308</td>
<td>0.323</td>
<td>+0.9</td>
<td>+4.9</td>
</tr>
<tr>
<td>8</td>
<td>0.350</td>
<td>0.343</td>
<td>0.352</td>
<td>+1.4</td>
<td>+2.6</td>
</tr>
</tbody>
</table>

MNZ=CombMNZ(Fox, Shaw, Lee et al)
COND=Condorcet(Aslam, Montague)
experiments – metasearch – TREC8

![Graph showing metasearch performance](image)

- **Hedge**
- **TREC8**

**Graph Details**
- **Y-axis**: MAP (Mean Average Precision)
  - 0.35 to 0.55
- **X-axis**: # docs judged
  - 0 to 500

Legend:
- Hedge
- CombMNZ
- Best system
metasearch – TREC 3,5,6,7
a powerful machine learning approach
  - Hedge = AdaBoost core

works [usually] better than anything else we’ve seen

true, it uses feedback
  - but without feedback there are provable limitations
Why hedge [schapire, freund ’96]

\[
\sum_{i=1}^{N} w_i^{T+1} \leq \left( \sum_{i=1}^{N} w_i^T \right) \left( 1 - (1 - \beta) p^T \cdot l^T \right) \leq \ldots
\]

\[
\ldots \leq \exp(- (1 - \beta) \sum_{t=1}^{T} p^t \cdot l^t)
\]

hedge loss at episode \( T = p^T \cdot l^T \)

cummulat loss \( L_{HEDGE} \)

\[
L_{HEDGE} \leq \frac{\ln \left( \frac{1}{\beta} \right) L_{SYSTEM} + \ln N}{1 - \beta}
\]

\( L_{Hedge}(\beta) \)

\[
\sum_{i=1}^{N} w_i^{T+1}
\]
AdaBoost – distribution update

\[ h_t : X \rightarrow \{-1,1\} \]

\[ \varepsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i] \]

\[ \alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right) > 0 \]

\[ D_{t+1} = \frac{D_t}{Z_t} \cdot e^{\alpha_t} \]

\[ D_{t+1} = \frac{D_t}{Z_t} \cdot e^{-\alpha_t} \]

- “neutralize” the last weak hypothesis
problem setup

- On the same query
- Set of underlying systems
- User feedback
- Goal
  - Find relevant documents
  - Produce online metasearch lists
  - Perform online system evaluation
- We are looking for an adaptive approach
our unified model
\[ \text{LOSS}(d, s) = \text{label}(d) \times \left( \frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{Z} \right) \approx \text{label}(d) \times \ln \frac{Z}{r} \]

\[ \text{TP}(s) = \text{total precision of } s = \text{average precision at all ranks} \]

FACT: \[ \sum_{\text{all docs}} \text{LOSS}(d, s) = C \times (Z - 2 \times \text{TP}(s)) \]

- Average the precision at ALL ranks
- Normalize so ideal system gets TP=1
- Math is more simple
- select $d^* = \arg \max_{\tilde{d} \in M} \sum_{s=1}^{t-1} w_s * \text{LOSS}(d, s | \tilde{d} = \text{NR})$

If RELEVANT, then a NR in top ranks of system lists

Naturally "want" top ranks

If NON RELEVANT, then a NR in top ranks of the system lists

If RELEVANT, bingo.

Hedge pooling – howto
system evaluation – howto

- assume all docs not judged (so many ?) to be NON RELEVANT
- compute AvegPrecision for every system
- one (or few) very good systems – use small $\beta$
Compute “pooling value” for each doc
  - Instead of “select the top doc” for pooling
  - do “select the top 1000 doc” for metasearch

almost 1000 – docs already pooled are automatically in top of metasearch list
experiments

- **TREC**
  - ~100 systems
  - 50 queries each competition
- **Use TREC qrels as user feedback**
  - incomplete feedback
- **For comparison with depth-pooling we use average number of pools (over queries)**
experiments - relevant docs found

Hedge

Cormack

TREC

depth pooling
experiments - system evaluation

- SYSTEM EVALUATION: Hedge-40 TREC8
- SYSTEM EVALUATION: depth-1 pooling TREC8
system evaluation – kendall’s tau

![Graph showing Kendall's tau for system ordering, Hedge, Depth-n, and TREC8. The x-axis represents the number of documents judged, and the y-axis represents Kendall's tau. The graph compares the performance of Hedge and Depth-n methods.]
### metasearch - no feedback (yet)

<table>
<thead>
<tr>
<th>TREC</th>
<th>MNZ</th>
<th>COND</th>
<th>Hedge-0</th>
<th>$%MNZ$</th>
<th>$%COND$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.423</td>
<td>0.403</td>
<td>0.418</td>
<td>$-1.2$</td>
<td>$+3.7$</td>
</tr>
<tr>
<td>5</td>
<td>0.294</td>
<td>0.307</td>
<td>0.309</td>
<td>$+5.1$</td>
<td>$+0.6$</td>
</tr>
<tr>
<td>6</td>
<td>0.341</td>
<td>0.315</td>
<td>0.345</td>
<td>$+1.2$</td>
<td>$+9.5$</td>
</tr>
<tr>
<td>7</td>
<td>0.320</td>
<td>0.308</td>
<td>0.323</td>
<td>$+0.9$</td>
<td>$+4.9$</td>
</tr>
<tr>
<td>8</td>
<td>0.350</td>
<td>0.343</td>
<td>0.352</td>
<td>$+1.4$</td>
<td>$+2.6$</td>
</tr>
</tbody>
</table>

**no relevant judgements**
experiments – metasearch – TREC8

![Graph showing metasearch performance comparison between Hedge, CombMNZ, and the best system. The graph plots MAP against the number of docs judged.]
metasearch – TREC 3,5,6,7
A powerful machine learning approach
  - Hedge = AdaBoost core

Works [usually] better than anything else we’ve seen

True, it uses feedback
  - But without feedback there are provable limitations

It is missing a rigorous analysis
  - We are not very far away with that
  - Need a model assumption
**After conclusion – don’t read**

**TREC8 competition**
- 129 search engines (systems)
- ~1000 docs returned /query
- TREC Evaluation DEPTH 100
- Judge top 100 docs (each sys); use MAP to rank systems

**fuse the lists (metasearch)**

**Hedge**
- LOSS FUNCTION
- DOCUMENT SELECTION
- SYSTEM WEIGHTS
- JUDGE
- FEEDBACK
- MODIFY WEIGHTS

**KEY:**
- FIND RELEVANT DOCS
recall rate

Recall rate as a function of the number of documents judged.

- **Depth-n**
- **Hedge**

Observations at the following points:
- 10
- 15
- 20
system evaluation

"some" method evaluation vs. real evaluation
pooling - comparison with Cormack
motivation

online metasearch

best underlying system

metasearch

avg precision

feedback
Start with uniform distribution $D$

- Choose Allocation
- Update distribution
- Receive loss vector
- select $d_t = \arg \max_{d \text{ not labeled}} \left[ \sum_{s=1}^{N} w_s^{t-1} \times LOSS(d, s | d = NR) \right]$

Naturally “want” top ranks

If NON RELEVANT, then a NR in top ranks of the system lists

If RELEVANT, bingo.

$LOSS(d, s | d = NR) = \left( \frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{Z} \right) \approx \ln \frac{Z}{r}$
Before the next episode

- Compute “pooling value” for each doc
- Instead of “select the top doc” for pooling do “select the top 1000 doc” for metasearch
- almost 1000 – docs already pooled are automatically in top of metasearch list
Average the precision at ALL ranks
  ● Normalize so ideal system gets TP=1

math is more simple
Metas – trec5

Metasearch Performance  Hedge  TREC5

MAP

# docs judged
Metas – trec6

The diagram shows the metasearch performance of Hedge and TREC6 systems. The x-axis represents the number of documents judged, ranging from 0 to 500, while the y-axis represents MAP (mean average precision). The performance of Hedge is indicated by a solid line, while the best system is shown by a dashed line.
Metas – trec7

![Graph showing metasearch performance for Hedge and TREC7 datasets. The graph plots MAP against the number of docs judged.](image)
Metasearch Performance  Hedge  TREC3

MAP

# docs judged

Hedge
CombMNZ
best system

0.49
0.48
0.47
0.46
0.45
0.44
0.43
0.42
0  50   100  150  200  250  300  350  400  450  500