

The K-armed Dueling Bandits Problem

**Yisong Yue, Joseph Broder, Bobby Kleinberg,
Thorsten Joachims**

**Department of Computer Science
Cornell University**

Adaptive Information Systems

- **Retrieval Function: $f(q) \rightarrow r$**
 - Input: q (query)
 - Output: r (ranking by relevance)
- **Conventional Systems**
 - One-size-fits-all
 - Hand-tuned and static retrieval function
- **Room for Improvement**
 - Different users need different retrieval functions
 - Different collections need different retrieval functions
- **Machine Learning**
 - Learn improved retrieval functions

Google Search: svm - Microsoft Internet Explorer

Address: <http://www.google.com/search?sourceid=navclient&ie=UTF-8&oe=UTF-8&q=svm>

Google svm

Google Search

Web Images Groups Directory News

Searched the web for **svm**. Results 1 - 10 of about 329,000. Search took 0.29 seconds.

Categories: [Computers > Artificial Intelligence > Machine Learning](#)
[Computers > Artificial Intelligence > Neural Networks > Software](#)

Show stock quotes for SVM (ServiceMaster Company The)

Bienvenue sur [svm.vnunet.fr](#)! - [Translate this page]
... Les forums de SVM. Participez aux grands débats de la rédaction. De vous à vous. Les meilleures réponses sélectionnées sur le forum de SVM. ...
[svm.vnunet.fr](#) - 39k - Mar 1, 2004 - Cached - Similar pages

SVM-Light Support Vector Machine
SVM-Light Support Vector Machine. Hier finden Sie Informationen zu den folgenden Themen: Thorsten Joachims' SVM light, SVMlight, Support Vector ...
Description: Training software for large-scale SVMs. [Free for non-commercial use]
Category: [Computers > Artificial Intelligence > ... > Software](#)
[svmlight.joachims.org](#) - 3k - Mar 1, 2004 - Cached - Similar pages

Support Vector Machine
... Support Vector Machine. The most recent SVM light page can now be found at <http://svmlight.joachims.org/>. Older versions are still available from here. ...
[www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light.html](#) - 6k - Cached - Similar pages

ServiceMaster -- We Are Home
ServiceMaster Issues Information on Tax Treatment of Dividends. ServiceMaster Reports 2003 Fourth Quarter Revenues and Profits. ServiceMaster ...
[www.svm.com](#) - 13k - Mar 1, 2004 - Cached - Similar pages

Kernel Machines
Description: A collection of information on kernel based methods, including support vector machines, Gaussian ...
Category: [Computers > Artificial Intelligence > Support Vector Machines](#)
[www.kernel-machines.org](#) - 1k - Cached - Similar pages

SVM Application List
SVM Application List. This list of Support Vector Machine applications grows thanks to visitors like you who ADD new entries. ... [svm learning](#) ...

Motivation and Outline

- **Setup**
 - Corpus of documents [*known*]
 - Distribution of users and/or queries on corpus [*unknown*]
 - Set of retrieval functions $\{f_1, \dots, f_K\}$ [*design choice*]
 - Each retrieval function f_i has utility $U(f_i)$ [*unknown*]
- **Question 1: How can one measure utility?**
 - Cardinal vs. ordinal utility measurements
 - Eliciting implicit feedback through interactive experiments
- **Question 2: How to efficiently find f_i with max utility?**
 - Efficiently \rightarrow minimizing regret + computationally efficient
 - Minimize exposure to suboptimal results during learning
 - Dueling Bandits Problem with efficient algorithm

Approaches to Implicit Utility Elicitation

- **Approach 1: Absolute Metrics (cardinal)**
 - Do metrics derived from observed user behavior provide absolute feedback about retrieval quality of f ?
 - For example:
 - $U(f) \sim \text{numClicks}(f)$
 - $U(f) \sim 1/\text{abandonment}(f)$
- **Approach 2: Paired Comparison Tests (ordinal)**
 - Do paired comparison tests provide relative preferences between two retrieval functions f_1 and f_2 ?
 - For example:
 - $f_1 \succ f_2 \Leftrightarrow \text{pairedCompTest}(f_1, f_2) > 0$

Paired Comparisons: Balanced Interleaving

$(u=tj, q=\text{"svm"})$

$f_1(u, q) \rightarrow r_1$

$f_2(u, q) \rightarrow r_2$

1. Kernel Machines
<http://svm.first.gmd.de/>
2. Support Vector Machine
<http://jbolivar.freesevers.com/>
3. An Introduction to Support Vector Machines
<http://www.support-vector.net/>
4. Archives of SUPPORT-VECTOR-MACHINES ...
<http://www.jiscmail.ac.uk/lists/SUPPORT...>
5. SVM-Light Support Vector Machine
<http://ais.gmd.de/~thorsten/svm light/>

1. Kernel Machines
<http://svm.first.gmd.de/>
2. SVM-Light Support Vector Machine
<http://ais.gmd.de/~thorsten/svm light/>
3. Support Vector Machine and Kernel ... References
<http://svm.research.bell-labs.com/SVMrefs.html>
4. Lucent Technologies: SVM demo applet
<http://svm.research.bell-labs.com/SVT/SVMsvt.html>
5. Royal Holloway Support Vector Machine
<http://svm.dcs.rhbnc.ac.uk>

Interleaving(r_1, r_2)

1. Kernel Machines 1
<http://svm.first.gmd.de/>
2. Support Vector Machine 2
<http://jbolivar.freesevers.com/>
3. SVM-Light Support Vector Machine 2
<http://ais.gmd.de/~thorsten/svm light/>
4. An Introduction to Support Vector Machines 3
<http://www.support-vector.net/>
5. Support Vector Machine and Kernel ... References 3
<http://svm.research.bell-labs.com/SVMrefs.html>
6. Archives of SUPPORT-VECTOR-MACHINES ... 4
<http://www.jiscmail.ac.uk/lists/SUPPORT...>
7. Lucent Technologies: SVM demo applet 4
<http://svm.research.bell-labs.com/SVT/SVMsvt.html>

Invariant:

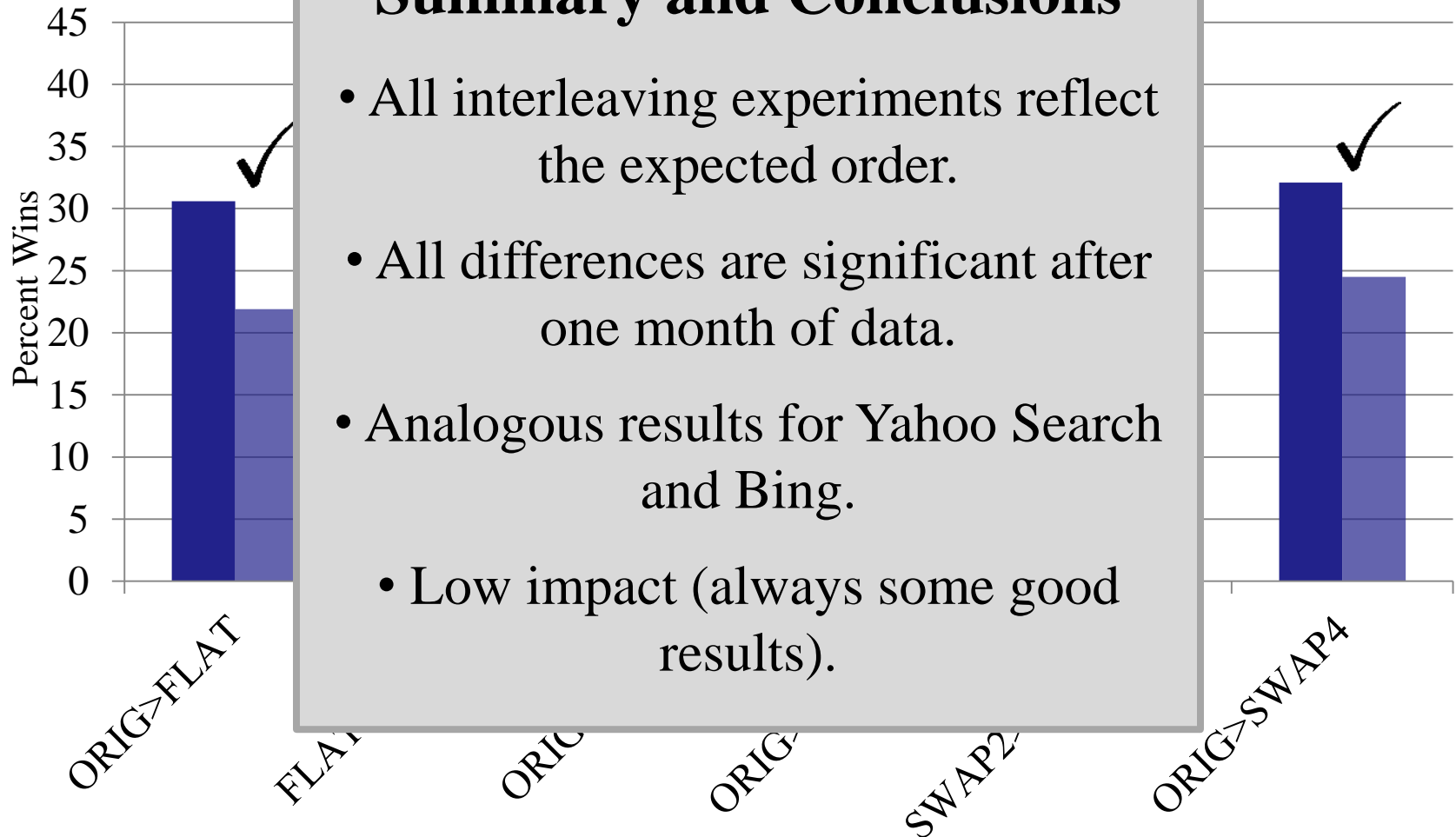
For all k , top k of balanced interleaving is union of top k_1 of r_1 and top k_2 of r_2 with $k_1 = k_2 \pm 1$.

Interpretation: $(r_1 \succ r_2) \leftrightarrow \text{clicks}(\text{topk}(r_1)) > \text{clicks}(\text{topk}(r_2))$

Balanced Interleaving: Results

Paired Comparison Tests: Summary and Conclusions

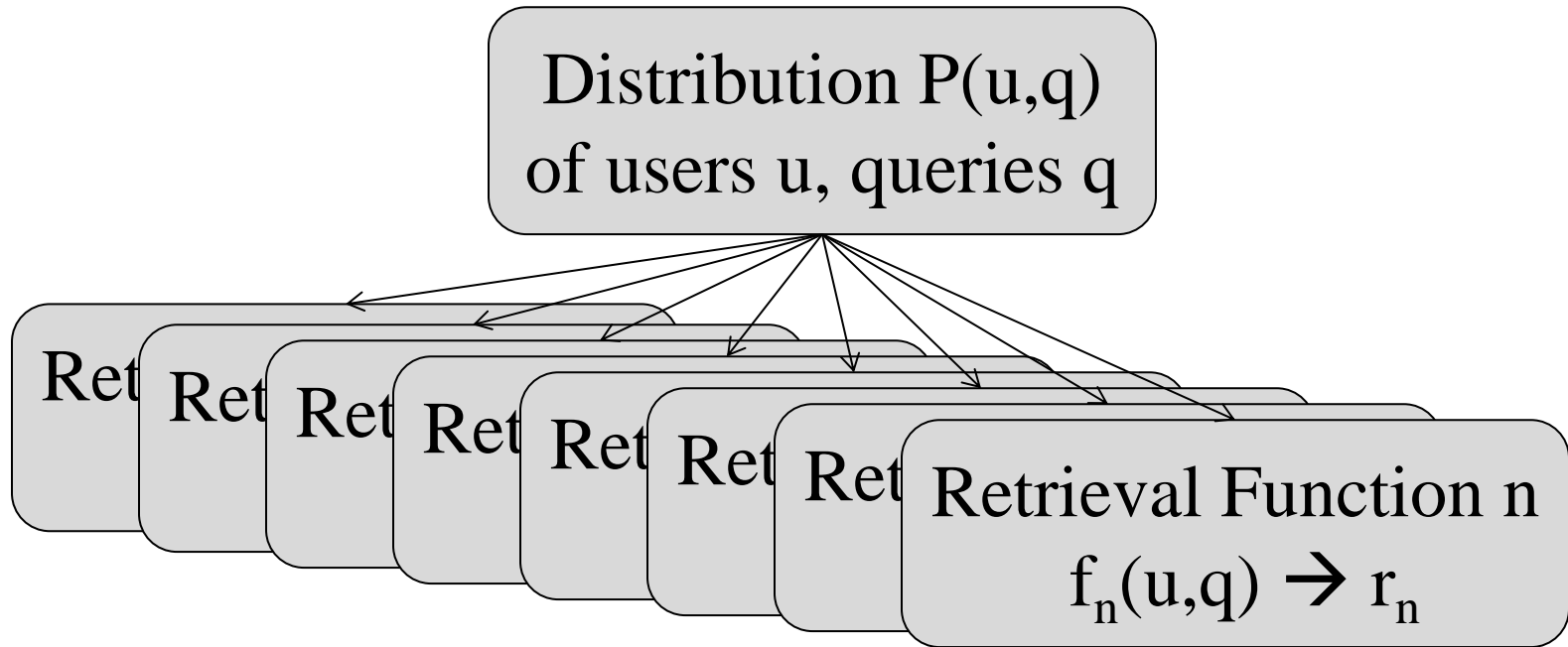
- All interleaving experiments reflect the expected order.
- All differences are significant after one month of data.
- Analogous results for Yahoo Search and Bing.
- Low impact (always some good results).



Motivation and Outline

- **Setup**
 - Corpus of documents [*known*]
 - Distribution of users and/or queries on corpus [*unknown*]
 - Set of retrieval functions $\{f_1, \dots, f_K\}$ [*design choice*]
 - Each retrieval function f_i has utility $U(f_i)$ [*unknown*]
- **Question 1: How can one measure utility?**
 - Cardinal vs. ordinal utility measurements
 - Eliciting implicit feedback through interactive experiments
- **Question 2: How to efficiently find f_i with max utility?**
 - Efficiently \rightarrow minimizing regret + computationally efficient
 - Minimize exposure to suboptimal results during learning
 - Dueling Bandits Problem with efficient algorithm

Evaluating Many Retrieval Functions



Task:

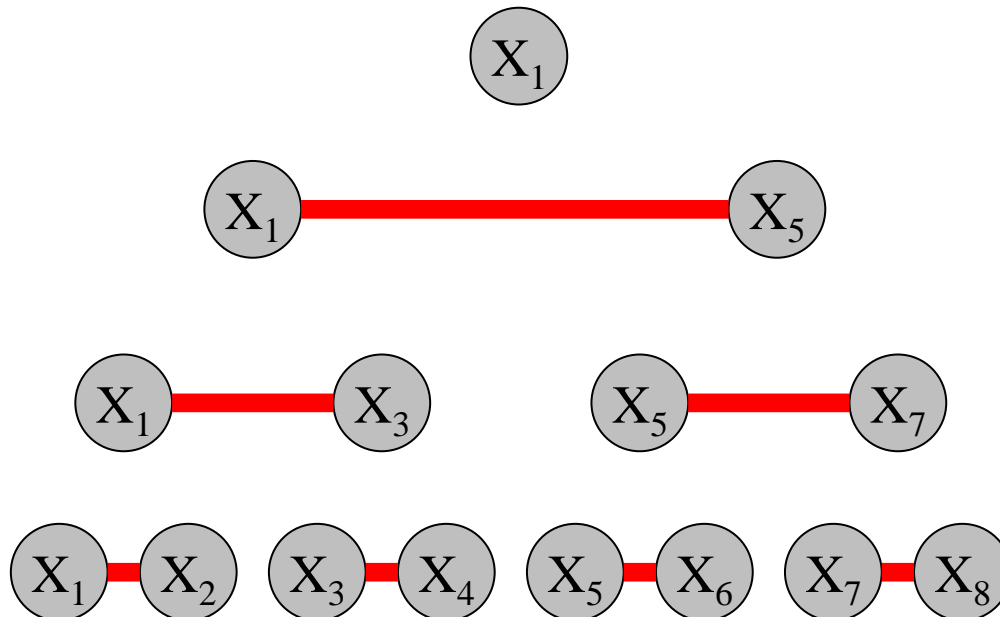
Find $f^* \in F$ that gives best retrieval quality over $P(u,q)$?

Tournament

- **Can you design a tournament that reliably identifies the correct winner?**

→ Noisy Sorting/Max Algorithms:

- [Feige et al.]: Triangle Tournament Heap $O(n/\epsilon^2 \log(1/\delta))$ with prob $1-\delta$
- [Adler et al., Karp & Kleinberg]: optimal under weaker assumptions



Problem: Learning on Operational System

- **Example:**

- 4 retrieval functions: $B > G \gg Y > A$
- 10 possible pairs for interactive experiment
 - (B,G) \rightarrow low cost to user
 - (B,Y) \rightarrow medium cost to user
 - (Y,A) \rightarrow high cost to user
 - (B,B) \rightarrow zero cost to user
 - ...

- **Minimizing Regret**

- Algorithm gets to decide on the sequence of pairwise tests
- Don't present "bad" pairs more often than necessary
- Trade off (long term) informativeness and (short term) cost

\rightarrow Dueling Bandits Problem

Regret for the Dueling Bandits Problem

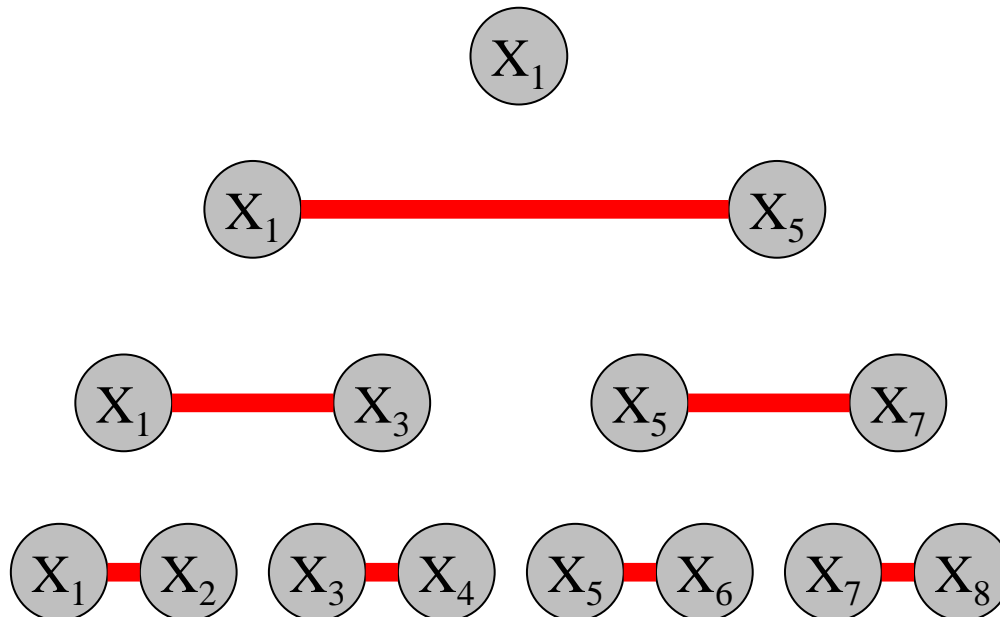
- **Given:**
 - A finite set H of candidate retrieval functions $f_1 \dots f_K$
 - A pairwise comparison test $f \succ f'$ on H with $P(f \succ f')$
- **Regret:**
 - $R(A) = \sum_{t=1..T} [P(f^* \succ f_t) + P(f^* \succ f'_t) - 1]$
 - f^* : best retrieval function in hindsight (assume single f^* exists)
 - (f, f') : retrieval functions tested at time t

Example:

Time Step:	t_1	t_2	...	T
Comparison:	$(f_9, f_{12}) \rightarrow f_9$	$(f_5, f_9) \rightarrow f_5$		$(f_1, f_3) \rightarrow f_3$
Regret:	$P(f^* \succ f_9) + P(f^* \succ f_{12}) - 1$ $= 0.9$	$P(f^* \succ f_5) + P(f^* \succ f_9) - 1$ $= 0.78$		$= 0.01$

Tournament

- **Can you design a tournament that has low regret?**
→ Don't know!



Algorithm: Interleaved Filter 1

MATCH

- Algorithm

InterleavedFilter1($T, W = \{f_1 \dots f_K\}$)

- Pick random f^* from W
- $\delta = 1/(TK^2)$

- WHILE $|W| > 1$

- FOR $f \in W$ DO

- » duel(f^*, f)

- » update P_f

- $t = t + 1$

- $c_t = (\log(1/\delta)/t)^{0.5}$

- Remove all f from W with $P_f < 0.5 - c_t$ [WORSE WITH PROB $1 - \delta$]

- IF there exists f^{**} with $P_{f^{**}} > 0.5 + c_t$ [BETTER WITH PROB $1 - \delta$]

- » Remove f^* from W

- » $f^* = f^{**}$; $t = 0$

- UNTIL T : duel(f^*, f^*)

EXPLORE
+
EXPLOIT

EXPLOIT

ROUND

f_1	f_2	$f^* = f_3$	f_4	f_5
0/0	0/0		0/0	0/0
f_1	f_2	$f^* = f_3$	f_4	f_5
8/2	7/3		4/6	1/9
f_1	f_2	$f^* = f_3$	f_4	
13/2	11/4		7/8	XX
$f^* = f_1$	f_2		f_4	
0/0	0/0	XX	0/0	XX

IF1: Main Result

- **Theorem:** The expected regret of IF1 is

$$E[R_T] = O\left(\frac{K \log K}{\epsilon_{1,2}} \log T\right)$$

where $\epsilon_{12} = P(f_1 \succ f_2) - 0.5$ and K is the number of bandits.

- **Assumptions:**
 - Strong Stochastic Transitivity: $\epsilon_{i,k} \geq \max\{\epsilon_{i,j}, \epsilon_{j,k}\}$
 - Stochastic Triangle Inequality: $\epsilon_{i,k} \leq \epsilon_{i,j} + \epsilon_{j,k}$
 - ϵ -winner exists

Assumptions

- **Preference Relation:** $f_i \succ f_j \Leftrightarrow P(f_i \text{ beats } f_j) = 0.5 + \varepsilon_{i,j} > 0.5$

- **Weak Stochastic Transitivity:** $f_i \succ f_j$ and $f_j \succ f_k \rightarrow f_i \succ f_k$

$$f_1 \succ f_2 \succ f_3 \succ f_4 \succ f_5 \succ f_6 \succ \dots \succ f_K$$

- **Strong Stochastic Transitivity:** $\varepsilon_{i,k} \geq \max\{\varepsilon_{i,j}, \varepsilon_{j,k}\}$

$$\varepsilon_{1,4} \geq \varepsilon_{2,4} \geq \varepsilon_{3,4} \geq 0.5 \geq \varepsilon_{5,4} \geq \varepsilon_{6,4} \geq \dots \geq \varepsilon_{K,4}$$

- **Stochastic Triangle Inequality:** $f_i \succ f_j \succ f_k \rightarrow \varepsilon_{i,k} \leq \varepsilon_{i,j} + \varepsilon_{j,k}$

$$\varepsilon_{1,2} = 0.01 \text{ and } \varepsilon_{2,3} = 0.01 \rightarrow \varepsilon_{1,3} \leq 0.02$$

- **ε -Winner exists:** $\varepsilon = \max_i\{P(f_1 \text{ beats } f_i) - 0.5\} = \varepsilon_{1,2} > 0$

IF1: Proof Outline

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

1. The probability that IF1 returns suboptimal bandit is less than 1/T

→ a) Probability that a match has wrong winner is at most $\delta = 1/(TK^2)$.

→ b) Upper bound on the number of matches: K^2

f_1	...	f_{K-2}	f_{K-1}	f_K
0/0	0/0	0/0	0/0	
1	...	f_{K-2}	f_{K-1}	f_K
0/0	0/0	0/0	0/0	XX
f_1	...	f_{K-2}	f_{K-1}	f_K
0/0	0/0	0/0	XX	XX

2. Bound expected regret $E[R_T^{IF1}]$ of IF1

- Bound number of duels in a match: $O(1/\epsilon^2)$
- Bound regret per match
- Bound the number of rounds before IF1 terminates

Lemma 1a: Probability that a Match has Wrong Winner is at most $\delta=1/(T K^2)$

- **Proof:**

- Reminder: Confidence interval $c_t=(\log(1/\delta)/t)^{0.5}$
- If we declare the wrong winner between f_i and f_j , then observed P_t must have been outside confidence interval.
- $P(|P_t - E[P_t]| \geq c_t) \leq 2 \exp(-2 t c_t^2) = 2\delta^2 = 2/(T^2 K^4)$
- Union bound over all time steps: $2T/(T^2 K^4) \leq 1/(T K^2) = \delta$

IF1: Proof Outline

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

1. The probability that IF1 returns suboptimal bandit is less than $1/T$ ✓

a) Probability that a match has wrong winner is at most $\delta = 1/(TK^2)$. ✓

b) Upper bound on the number of matches: K^2 ✓

2. Bound expected regret $E[R_T^{IF1}]$ of IF1

→ a) Bound number of duels in a match

b) Bound regret per match

c) Bound the number of rounds before IF1 terminates

Lemma 2a:

Bound Number of Duels in a Match

- Consider: match between f_i and f_j with $P(f_i \text{ beats } f_j) = 0.5 + \varepsilon_{i,j}$
- If match is t duels long, then $P_t - c_t \leq 0.5$, otherwise the match would terminate.
- $P(n > t) \leq P(P_t - c_t \leq 0.5) = P(E[P_t] - P_t \geq \varepsilon_{i,j} - c_t)$
- For any $m \geq 4$ and $t = (m \log(TK^2) / \varepsilon_{i,j}^2)$, we have $c_t \leq 0.5 \varepsilon_{i,j}^2$.
- Hoeffding bound $\rightarrow O(1/\varepsilon_{i,j}^2 \log(TK))$ whp

IF1: Proof Outline

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

1. The probability that IF1 returns suboptimal bandit is less than $1/T$ ✓

a) Probability that a match has wrong winner is at most $\delta = 1/(TK^2)$. ✓

b) Upper bound on the number of matches: K^2 ✓

2. Bound expected regret $E[R_T^{IF1}]$ of IF1

a) Bound number of duels in a match: $O(1/\varepsilon_{i,j}^2 \log(TK))$ whp ✓

→ b) Bound regret per match

c) Bound the number of rounds before IF1 terminates

Lemma 2b: Bound Regret per Match

- Proof:**

f_1	...	$f^*=f_j$...	f_K
0/0	0/0		0/0	0/0

- Let current incumbent $f^*=f_j$:

- Note: no match involving f_j is longer than $O(1/\varepsilon_{1,j}^2 \log(TK))$ whp (Lemma 2a)

- Each $\text{duel}(f_i, f_j)$ incurs $(\varepsilon_{1,j} + \varepsilon_{1,i})$ regret:

$f_1 \quad f_i \quad f_j$

- Case $f_i \succ f_j$: Then $\varepsilon_{1,j} + \varepsilon_{1,i} \leq 2 \varepsilon_{1,j}$ (SST) and regret is bounded $2\varepsilon_{1,j} O(1/\varepsilon_{1,j}^2 \log(TK)) = O(1/\varepsilon_{1,j} \log(TK)) \leq O(1/\varepsilon_{1,2} \log(TK))$

$f_1 \quad f_j \quad f_i$

- Case $f_i \prec f_j$ and $\varepsilon_{j,i} \leq \varepsilon_{1,j}$: Then $\varepsilon_{1,j} + \varepsilon_{1,i} \leq \varepsilon_{1,j} + \varepsilon_{1,j} + \varepsilon_{j,i} \leq 3 \varepsilon_{1,j}$ due to STI. $3\varepsilon_{1,j} O(1/\varepsilon_{1,j}^2 \log(TK)) = O(1/\varepsilon_{1,j} \log(TK)) \leq O(1/\varepsilon_{1,2} \log(TK))$

$f_1 \quad f_j \quad f_i$

- Case $f_i \prec f_j$ and $\varepsilon_{j,i} > \varepsilon_{1,j}$: Then $\varepsilon_{1,j} + \varepsilon_{1,i} \leq \varepsilon_{1,j} + \varepsilon_{1,j} + \varepsilon_{j,i} \leq 3 \varepsilon_{j,i}$ at most $O(1/\varepsilon_{j,i} \log(TK))$ duels. $3\varepsilon_{j,i} O(1/\varepsilon_{j,i}^2 \log(TK)) = O(1/\varepsilon_{1,j} \log(TK)) \leq O(1/\varepsilon_{1,2} \log(TK))$

IF1: Proof Outline

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

1. The probability that IF1 returns suboptimal bandit is less than $1/T$ ✓

a) Probability that a match has wrong winner is at most $\delta = 1/(TK^2)$. ✓

b) Upper bound on the number of matches: K^2 ✓

2. Bound expected regret $E[R_T^{IF1}]$ of IF1

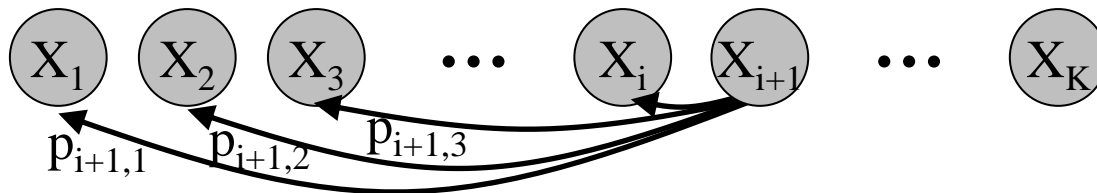
a) Bound number of duels in a match: $O(1/\varepsilon_{i,j}^2 \log(TK))$ whp ✓

b) Bound regret per match: $O(1/\varepsilon_{1,2} \log(TK))$ whp ✓

→ c) Bound the number of rounds before IF1 terminates

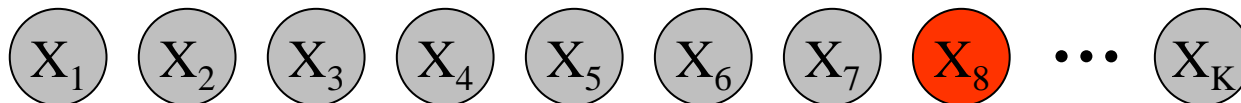
Lemma 2c: Bound the Number of Rounds before IF1 Terminates

- Random walk: $X_i=1$ if f_i becomes incumbent, $X_i=0$ else



→ $\sum X_i$ = number of steps in random walk = number of rounds

- Note: If IF1 does not make a mistake, then only forward steps.
- Strong Stochastic Transitivity: $\forall i: p_{i+1,1} \geq p_{i+1,2} \geq \dots \geq p_{i+1,i}$
 → Worst case: $p_{i+1,1} = p_{i+1,2} = \dots = p_{i+1,i} = 1/i$



- $\sum X_i = O(\log K)$ rounds whp

IF1: Proof Outline

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

1. **Theorem:** IF1 incurs expected regret bounded by

a)
$$E(R_T) \leq O\left(\frac{K \log K}{\epsilon_{1,2}} \log T\right)$$

b)

2. **Bound expected regret $E[R_T^{IF1}]$ of IF1**

a) Bound number of duels in a match: $O(1/\epsilon^2_{i,j} \log(TK))$ whp ✓

b) Bound regret per match: $O(1/\epsilon_{1,2} \log(TK))$ whp ✓

c) Bound rounds before IF1 terminates: $O(\log K)$ whp ✓

Lower Bound

- **Theorem:** Any algorithm for the dueling bandits problem has regret

$$R_T \leq \Omega \left(\frac{K}{\epsilon_{1,2}} \log T \right)$$

- **Proof:** [Karp/Kleinberg/07][Kleinberg/etal/07]
- **Intuition:**
 - Magically guess the best bandit, just verify guess
 - Worst case: $\forall f_i \succ f_j: P(f_i \succ f_j) = 0.5 + \epsilon$
 - Lemma 2a: Need $O(1/\epsilon^2 \log T)$ duels to get $1 - 1/T$ confidence.

Algorithm: Interleaved Filter 2

- Algorithm**

InterleavedFilter1($T, W = \{f_1 \dots f_K\}$)

- Pick random f^* from W
- $\delta = 1/(TK^2)$
- WHILE $|W| > 1$
 - FOR $b \in W$ DO
 - » $\text{duel}(f^*, f)$
 - » update P_f
 - $t = t + 1$
 - $c_t = (\log(1/\delta)/t)^{0.5}$
 - Remove all f from W with $P_f < 0.5 - c_t$ [WORSE WITH PROB $1 - \delta$]
 - IF there exists f^{**} with $P_{f^{**}} > 0.5 + c_t$ [BETTER WITH PROB $1 - \delta$]
 - » Remove f^* from W
 - » Remove all f from W that are empirically inferior to f^*
 - » $f^* = f^{**}$; $t = 0$
- UNTIL T : $\text{duel}(f^*, f^*)$

f_1	f_2	$f^* = f_3$	f_4	f_5
0/0	0/0		0/0	0/0
f_1	f_2	$f^* = f_3$	f_4	f_5
8/2	7/3		4/6	1/9
f_1	f_2	$f^* = f_3$	f_4	
13/2	11/4		7/8	XX
$f^* = f_1$	f_2		f_4	
0/0	0/0	XX	XX	XX

Why is it Safe to Remove Empirically Inferior Bandits?

- **Lemma:** Mistakenly pruning a bandit has probability at most $\delta=1/(T K^2)$.
- **Proof:**
 - Mistake: $f_p \succ f_w \succ f_i$ (pruned: f_p , winner: f_w , incumbent: f_i)
 - $B_{n,w,p}$: Given w is winner after n duels, f_p mistakenly pruned.
 - To show: $P(B_{n,w,p}) \leq 1-\delta$ for all n and w .
 - Suppose $P(b_w \succ b_i) = \alpha$ and given $B_{n,w,p} : P(b_p \succ b_i) \geq \alpha$.
→ $E(S_{w,i} + S_{i,p}) \leq n$.
 - Duels won $S_{w,i} - 0.5n < \sqrt{n \log(1/\delta)}$ and $S_{i,p} > 0.5n$
→ $S_{w,i} + S_{i,p} - n > \sqrt{n \log(1/\delta)}$
 - Hoeffding $P(S_{w,i} + S_{i,p} - n > \sqrt{n \log(1/\delta)}) \leq \delta$

Bound the Number of Matches of IF2

- **Lemma:** Assuming IF2 is mistake free, then it plays $O(K)$ matches in expectation.
- **Intuition:**



Regret Bound for IF2

$$E[R_T] \leq \left(1 - \frac{1}{T}\right) E[R_T^{IF1}] + \frac{1}{T} O(T) = O(E[R_T^{IF1}])$$

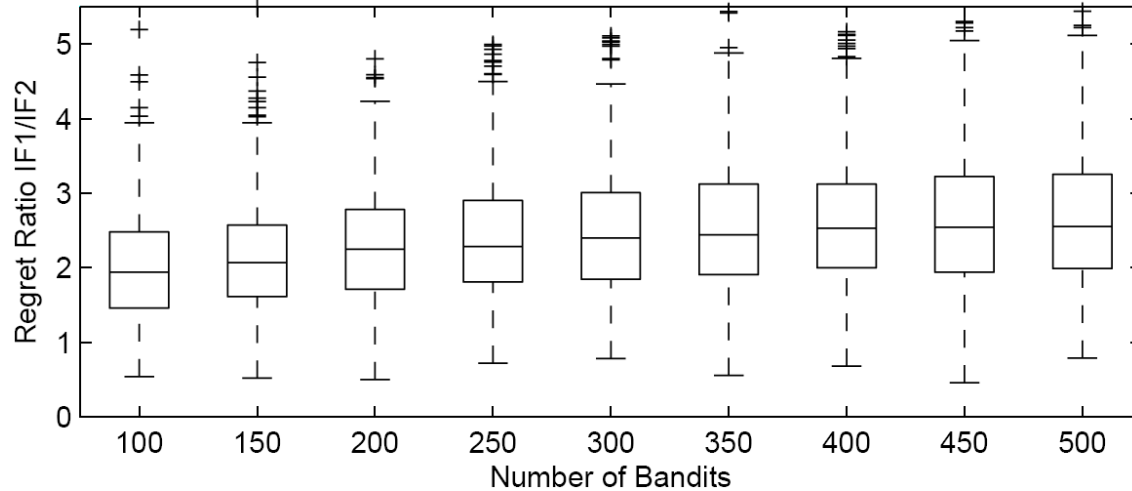
- **Lemma:** Mistakenly pruning a bandit has probability at most $\delta=1/(TK^2)$.
- **Lemma:** Assuming IF2 is mistake free, then it plays $O(K)$ matches in expectation.

Theorem: IF2 incurs expected regret bounded by

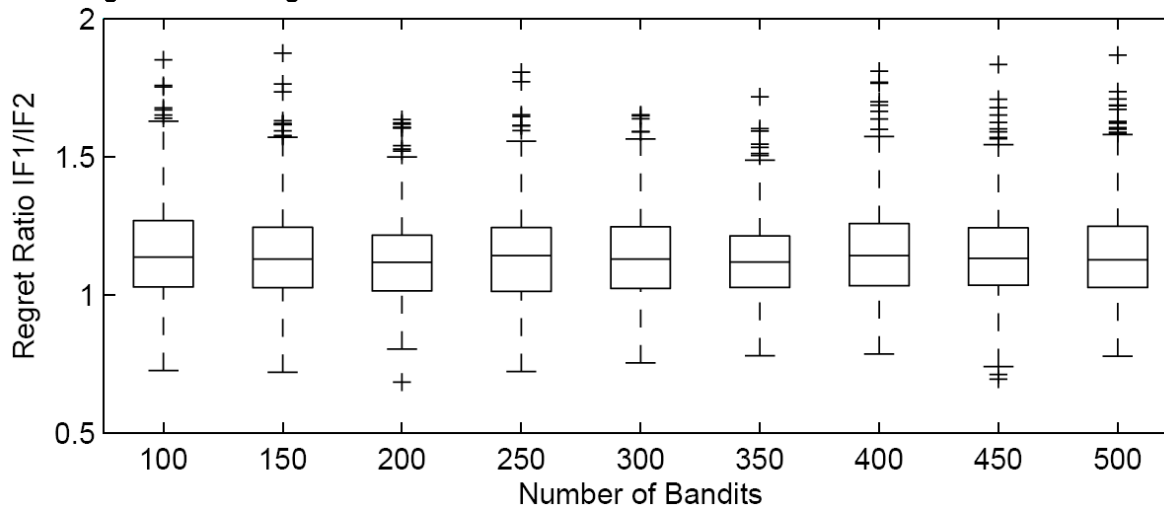
$$E(R_T) \leq O\left(\frac{K}{\epsilon_{1,2}} \log T\right)$$

Experiments: Synthetic Data

- **Lower-Bound data:** $\forall f_i \succ f_i: P(f_i \succ f_i)=0.5+\epsilon$

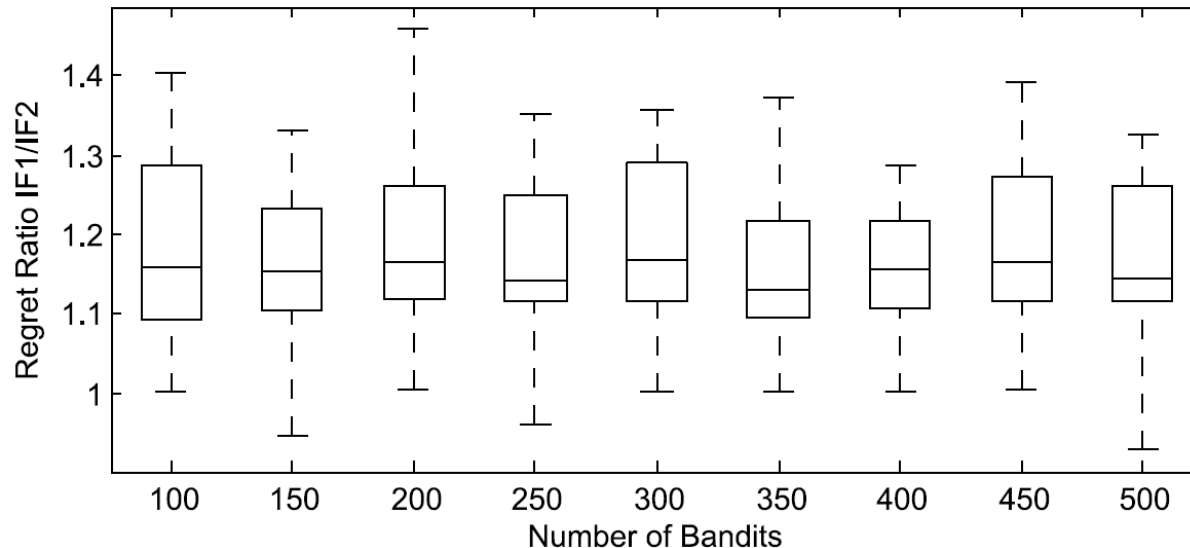


- **Bradley-Terry data**



Experiment: Simulated Web Search

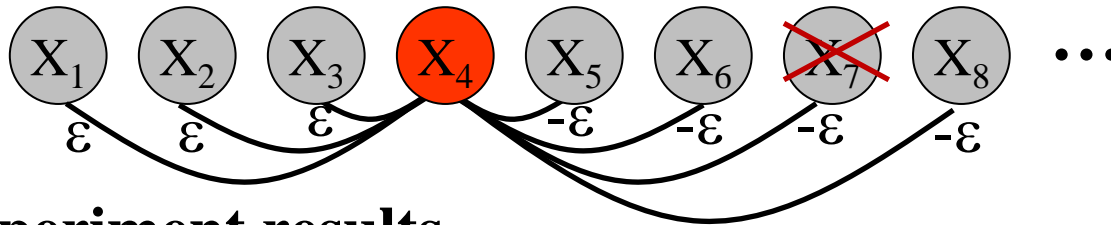
- **Microsoft Web Search Data (Chris Burges) with manual relevance assessment**
- **Feedback $f_i \succ f_j$:**
 - Draw query at random
 - Preference $f_i \succ f_j$ (probabilistically) based on NDCG difference of rankings produced by f_i and f_j



Why not a log-Gap?

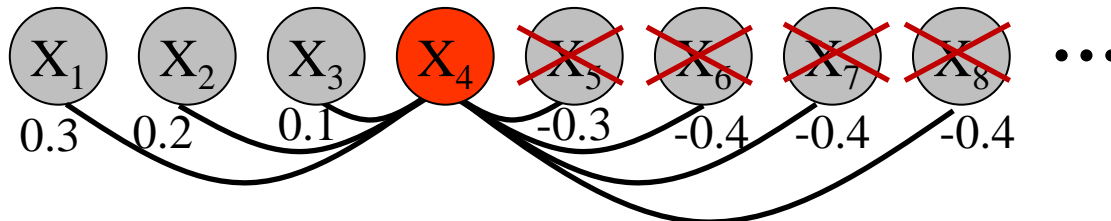
- **To achieve log-gap:**

- Log number of rounds need to be played
- Most inferior bandits must not get eliminated anyway without pruning.



- **Experiment results**

- Typically 2-4 rounds largely independent of number of bandits
- Many bandits much worse, so eliminated before round ends



Summary

- **Dueling Bandits Problem**

- Only ordinal information about payoffs
- Algorithms proposes two alternatives, user provides noisy preference.
- Preference can be interleaving, direct comparison, etc.

- **Interleaved Filter Algorithm**

- Regret based on win/loss against optimal bandit
- Strategy: keep incumbent, compare against others, prune inferior
- $O(K/\epsilon \log T)$ regret like for bandits with absolute feedback

- **Further Question**

- Beat-the-Mean-Bandit algorithm for K-armed dueling bandits problem [Yue & Joachims, 2011]
 - Lower variability
 - Relax strong stochastic transitivity
- Algorithm for finite and convex sets of bandit [Yue & Joachims, 2009]