Cornell University
Computer Science

The *K*-armed Dueling Bandits Problem

CS 6784 April 13th, 2010

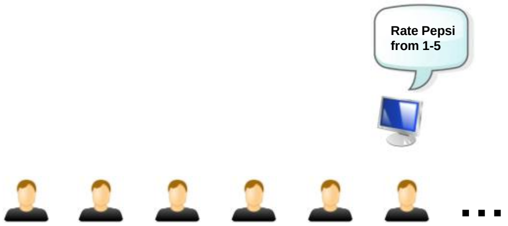
Yisong Yue
Cornell University

Joint work with:
Josef Broder, Robert Kleinberg, Thorsten Joachims

Online Learning

- Learn “on the fly”
  - Multi-armed Bandit Problem
- Broadly applicable
  - Many systems interact with environment
  - Can collect feedback, learn automatically
- How to analyze performance?
  - Utilities of strategies chosen vs best in hindsight
  - Also known as **“regret”**

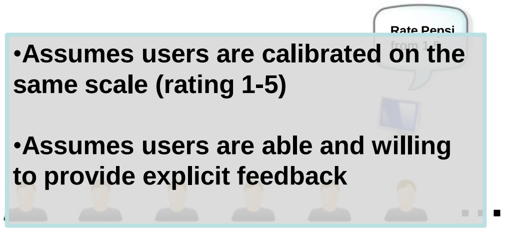
Absolute Explicit Feedback



Coke Avg: 2
Pepsi Avg: 2.75

Users are thirsty
Users drink & rate
Goal: Maximize satisfaction

Absolute Explicit Feedback



•Assumes users are calibrated on the same scale (rating 1-5)

•Assumes users are able and willing to provide explicit feedback

Coke Avg: 2
Pepsi Avg: 2.75

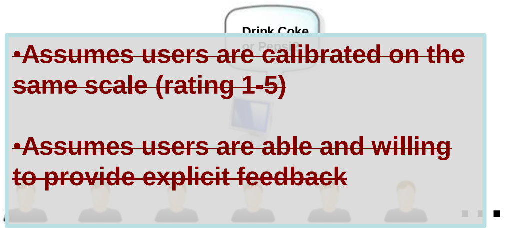
Users are thirsty
Users drink & rate
Goal: Maximize satisfaction

Relative Implicit Feedback

Coke Wins: 3
Pepsi Wins: 1

Users are thirsty
Observe users drinking
Goal: Maximize satisfaction

Relative Implicit Feedback



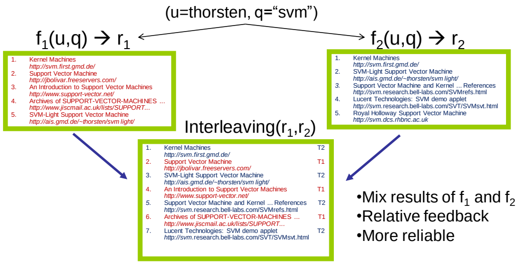
~~**•Assumes users are calibrated on the same scale (rating 1-5)**~~

~~**•Assumes users are able and willing to provide explicit feedback**~~

Coke Wins: 3
Pepsi Wins: 1

Users are thirsty
Observe users drinking
Goal: Maximize satisfaction

Team-Game Interleaving
(Comparison Oracle for Search Applications)



(u=thorsten, q="svm")

$f_1(u,q) \rightarrow r_1$

1. Kernel Machines
http://svm.first.gmd.de/
2. Support Vector Machine
http://jbolivar.freeservers.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/

$f_2(u,q) \rightarrow r_2$

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 T2
http://svm.first.gmd.de/
2. Support Vector Machine T1
http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine T2
http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines T1
http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References T2
http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ... T1
http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet T2
http://svm.research.bell-labs.com/SVT/SVMsvt.html

- Mix results of f_1 and f_2
- Relative feedback
- More reliable

Interpretation: ($r_1 > r_2$) ↔ clicks(T_1) > clicks(T_2)

[Radlinski, Kurup, Joachims, CIKM 2008]

Dueling Bandits Problem

- Given K bandits $b_1, \ldots, b_K$
- Each iteration: compare (duel) two bandits
  - E.g., interleaving two retrieval functions
- Comparison is noisy
  - Each comparison result independent
  - Comparison probabilities initially unknown
  - Comparison probabilities fixed over time
- Total preference ordering, initially unknown

Dueling Bandits Problem

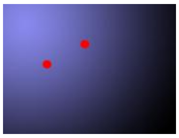
- Want to find best (or good) bandit
  - Similar to finding the max w/ noisy comparisons
  - Ours is a regret minimization setting
- Choose pair (b_t, b_t') to minimize regret:

$$R_T = \sum_{t=1}^{T} P(b^* > b_t) + P(b^* > b_t') - 1$$

- (% users who prefer best bandit over chosen ones)

$$R_T = \sum_{t=1}^{T} P(b^* > b_t) + P(b^* > b_t') - 1$$

- Example 1
  - P(f* > f) = 0.9
  - P(f* > f') = 0.8
  - Incurred Regret = 0.7



- Example 2
  - P(f* > f) = 0.7
  - P(f* > f') = 0.6
  - Incurred Regret = 0.3

- Example 3
  - P(f* > f) = 0.51
  - P(f* > f) = 0.55
  - Incurred Regret = 0.06

Assumptions

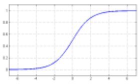
- P($b_i > b_j$) = ½ + ε_{ij} (distinguishability)
- **Strong Stochastic Transitivity**
  - For three bandits $b_i > b_j > b_k$:
  - Monotonicity property

$$\varepsilon_{ik} \geq \max\{\varepsilon_{ij}, \varepsilon_{jk}\}$$

- **Stochastic Triangle Inequality**
  - For three bandits $b_i > b_j > b_k$:
  - Diminishing returns property

$$\varepsilon_{ik} \leq \varepsilon_{ij} + \varepsilon_{jk}$$

- Satisfied by many standard models
  - E.g., Logistic / Bradley-Terry



Examples

	A	B	C	D
A	0	0.2	0.3	0.4
B	-0.2	0	0.1	0.3
C	-0.3	-0.1	0	0.1
D	-0.4	-0.3	-0.1	0

	A	B	C	D
A	0	0.2	0.3	**-0.1**
B	-0.2	0	0.1	0.3
C	-0.3	-0.1	0	0.1
D	**0.1**	-0.3	-0.1	0

	A	B	C	D
A	0	**0.2**	**0.4**	0.4
B	**-0.2**	0	**0.1**	0.3
C	**-0.4**	**-0.1**	0	0.1
D	-0.4	-0.3	-0.1	0

	A	B	C	D
A	0	**0.2**	**0.1**	0.4
B	**-0.2**	0	0.1	0.3
C	**-0.1**	-0.1	0	0.1
D	-0.4	-0.3	-0.1	0

P(A > B) = ½ + ε_{AB}

Explore then Exploit

- First explore
  - Try to gather as much information as possible
  - Accumulates regret based on which bandits we decide to compare
- Then exploit
  - We have a (good) guess as to which bandit best
  - Repeatedly compare that bandit with itself
    - (i.e., interleave that ranking with itself)

$$\boxed{R_T = \sum_{t=1}^{T} P(b^* > b_t) + P(b^* > b_t') - 1}$$

Goal

- An explore algorithm that finds the best bandit with probability at least 1-1/T

$$R_T = \sum_{t=1}^{T} P(b^* > b_t) + P(b^* > b_t') - 1$$

- Let R_E be the regret of running explore alg.

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

Goal

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

- Explore algorithm accumulates R_E = o(T)
- Average regret R_E/T converges to 0 as T grows
- Goal: $R_E = O\left(\frac{K}{\varepsilon} \log T\right)$ ε = min (ε_{12}, ε_{13}, … ε_{1K}) = ε_{12}

Mathematical Tools

- Union bound: $P\left(\bigcup_i A_i\right) \le \sum_i P(A_i)$
- Tail bound (Hoeffding): $P(S_n - E[S_n] \ge nL) \le e^{-2nL^2}$

  $X_1, \ldots, X_n$ (random variables between [0,1])

  $S_n = X_1 + \ldots + X_n$

- (probably the most useful slide in this lecture)

Comparing One Pair

- Comparisons are noisy
  - $P(b_i > b_j) = ½ + \varepsilon_{ij}$
  - (assume $\varepsilon_{ij} > 0$)
- How many comparisons are needed to confirm that $\varepsilon_{ij} > 0$ with confidence 1-δ?
- Can use Hoeffding bound to show $O\left(\frac{1}{\varepsilon_{ij}^2} \log \frac{1}{\delta}\right)$
  - (with high probability)

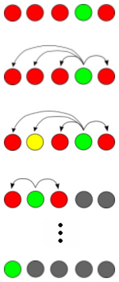
Naïve Approach

- In deterministic case, O(K) comparisons to find max
- Extend to noisy case:
  - Repeatedly compare until confident one is better
- Problem: comparing two awful (but similar) bandits
  - Waste comparisons to see which awful bandit is better
  - Incur high regret for each comparison
  - Also applies to elimination tournaments

$$\boxed{R_T = O\left(\frac{K}{\varepsilon^2} \log T\right)}$$

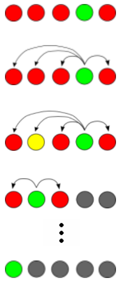
Interleaved Filter

- Choose candidate bandit at random
- Make noisy comparisons (Bernoulli trial) against all other bandits simultaneously
  - Maintain mean and confidence interval for each pair of bandits being compared
- …until another bandit is better
  - With confidence 1 – δ
- Repeat process with new candidate
  - (Remove all empirically worse bandits)
- Continue until 1 candidate left



Regret Analysis

- **Round:** all the time steps for a particular candidate bandit
  - Halts when better bandit found …
  - … with 1- δ confidence
  - Choose δ = 1/(TK^2)
- **Match:** all the comparisons between two bandits in a round
  - At most K matches in each round
  - Candidate plays one match against each remaining bandit



Per-Match Regret

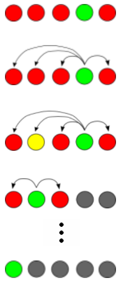
- Number of comparisons in match b_i vs b_j : $O\left(\frac{1}{\max\{\varepsilon_{1i}^2, \varepsilon_{ij}^2\}}\log T\right)$
  - $\varepsilon_{1i} > \varepsilon_{ij}$: round ends before concluding $b_i > b_j$
  - $\varepsilon_{1i} < \varepsilon_{ij}$: conclude $b_i > b_j$ before round ends, remove b_j
- Pay $\varepsilon_{1i} + \varepsilon_{1j}$ regret for each comparison
  - By triangle inequality $\varepsilon_{1i} + \varepsilon_{1j} \leq 2*\max\{\varepsilon_{1i}, \varepsilon_{ij}\}$
  - Thus by stochastic transitivity accumulated regret is

$$O\left(\frac{1}{\max\{\varepsilon_{1i}, \varepsilon_{ij}\}}\log T\right) \leq O\left(\frac{1}{\varepsilon}\log T\right)$$

$\varepsilon = \min(\varepsilon_{12}, \varepsilon_{13}, \ldots \varepsilon_{1K}) = \varepsilon_{12}$

Analyzing IF1

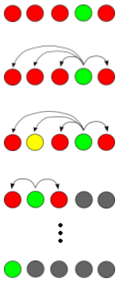
- At most K matches per round
- Regret per match: $O\left(\frac{1}{\varepsilon}\log T\right)$
- How many rounds?
  - Model the sequence of candidate bandits as a random walk
  - Can prove using tail bounds (Chernoff) that O(log K) rounds w.h.p.
- Total regret: $O\left(\frac{K\log K}{\varepsilon}\log T\right)$



Analyzing IF1

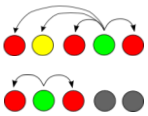
- How does IF1 avoid quadratic dependence on 1/ε?
  - Length of all matches in round bounded by length of match with winner
  - Does not waste time on “close” bandits
- But now has extra log K factor
  - Because there are log K rounds
  - Will fix this with IF2

$$O\left(\frac{K\log K}{\varepsilon}\log T\right)$$



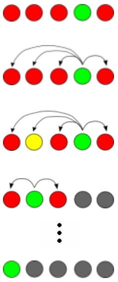
Removing Inferior Bandits

- At conclusion of each round
  - Remove any empirically worse bandits
- Intuition:
  - High confidence that winner is better than incumbent candidate
  - Empirically worse bandits cannot be “much better” than incumbent candidate
  - Can show via Hoeffding bound that winner is also better than empirically worse bandits with high confidence
  - Preserves 1-1/T confidence overall that we’ll find the best bandit



Analyzing IF2

- “Pruning” at the end of each round
  - Removes empirically inferior bandits
  - Even if not 1- δ confident
  - We still find best bandit w.p. 1-1/T
- How many pruned each round?
  - In expectation at least ¼ of remainder
- O(K) matches played in total.
- Expected Regret: $O\left(\frac{K}{\varepsilon}\log T\right)$

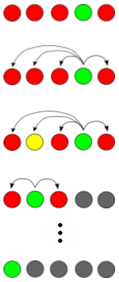


Analyzing IF2

- O(K) total matches
- Each match incurs regret $O\left(\frac{1}{\varepsilon}\log T\right)$
  - Depends on δ = $K^{-2}T^{-1}$
- Finds best bandit w.p. 1-1/T
- Expected regret:

$$E[R_T] = \left(1-\frac{1}{T}\right)O\left(\frac{K}{\varepsilon}\log T\right)+\frac{1}{T}O(T)$$

$$E[R_T] = O\left(\frac{K}{\varepsilon}\log T\right)$$



Limitations

- (Bandit ⇔ retrieval function)
- Ignores context
  - Maybe one is better for some queries/users but not others
- Assumes quality of retrieval functions are static
  - Maybe quality changes as users / documents change
- Inefficient for large K
  - Assume additional structure on for retrieval functions?
- Assumes strong stochastic transitivity
  - User preferences are probably not magnitude preserving