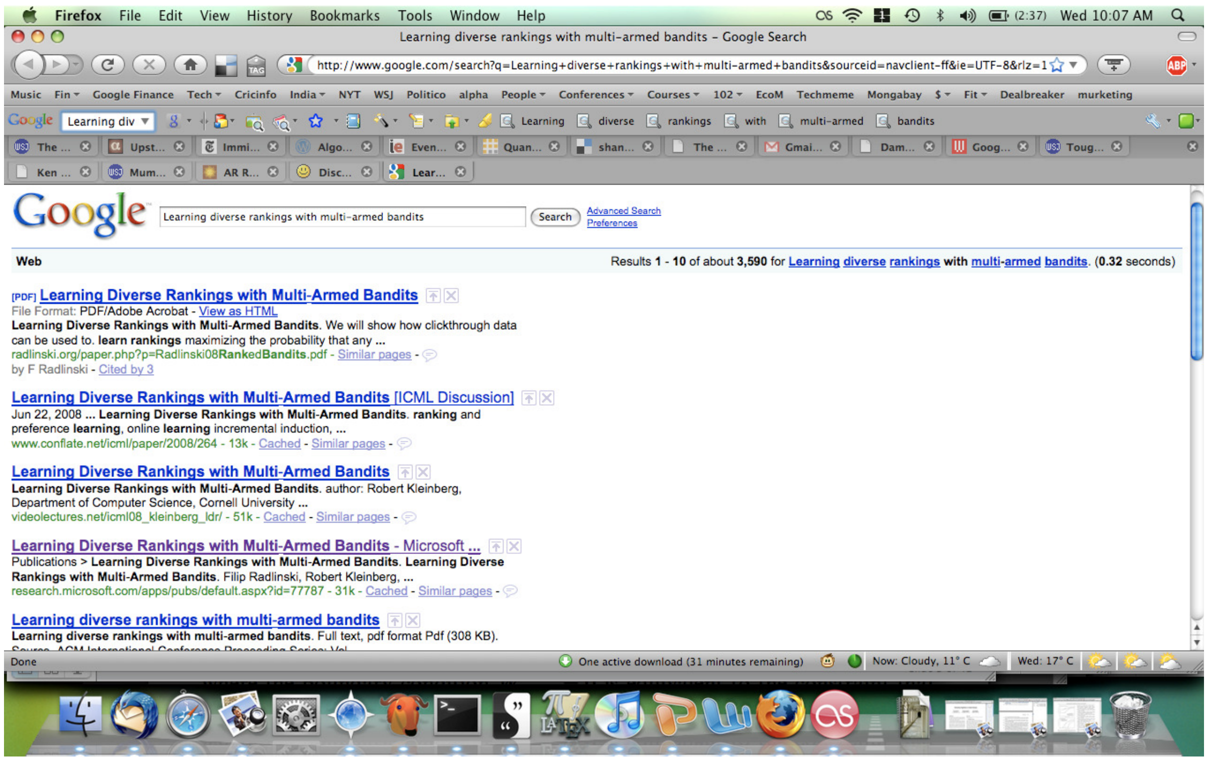
Learning diverse rankings with multi-armed bandits



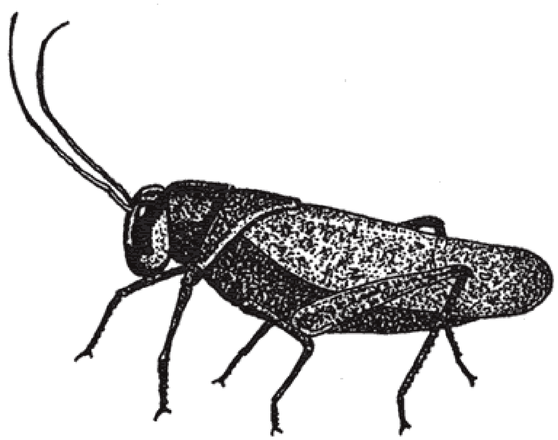
Radlinski, Kleinberg & Joachims. ICML '08

Overview

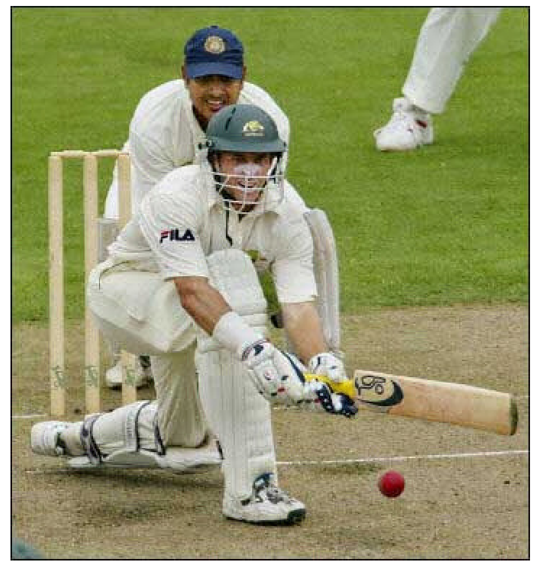
a) Problem of diverse rankings.
b) Solution approaches
c) Two possible candidates
d) Using multi-armed bandits
e) Theoretical analysis
f) Ranked explore and commit

Ranking search results on the Web

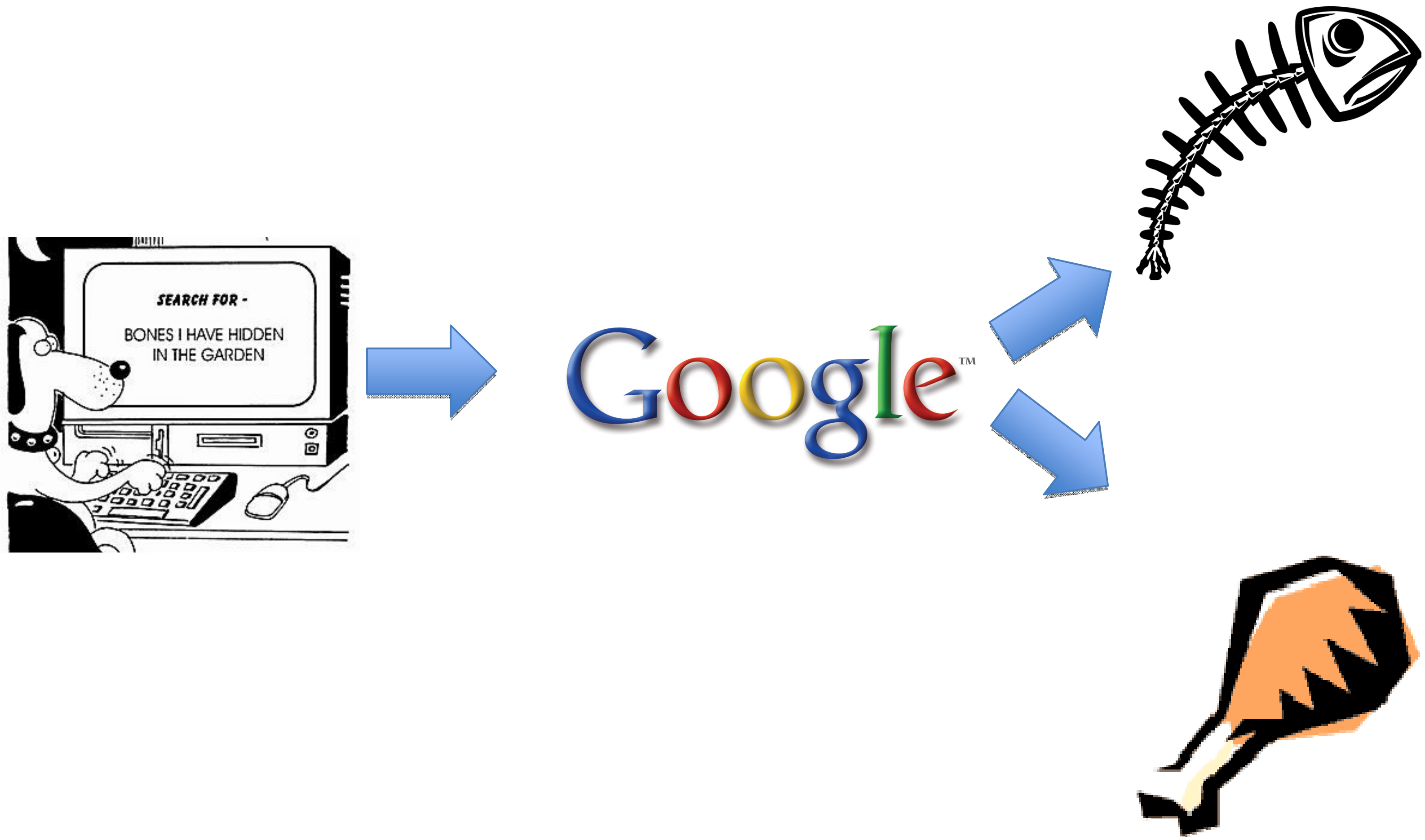
- A key metric used is "Relevance"
  - This can be different for different users
  - How to learn/infer the relevance?



OR

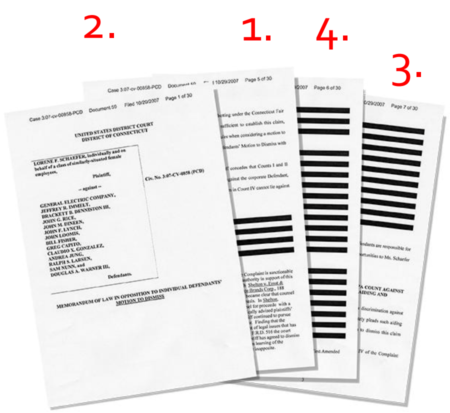


How to compute rankings?



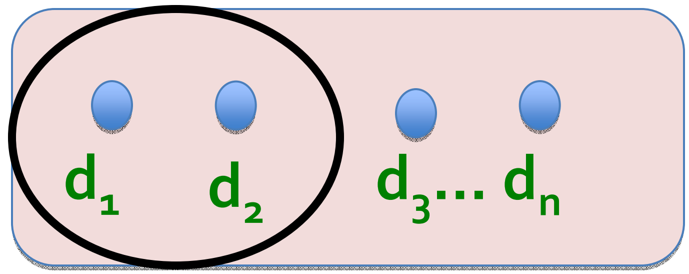
How to learn diverse rankings?

What should be used as training data?



Expert judgments

Using click-through data



Relevant set

$$\left\{ d_2 \quad d_1 \quad d_3 \right\}$$

Ordered set

Two approaches

- Ranked bandit algorithm
    - Think of the ranks as different copies of bandit algorithms running simultaneously
- Ranked Explore and Commit
    - Explores each document for a given rank and assigns rank based on user click data

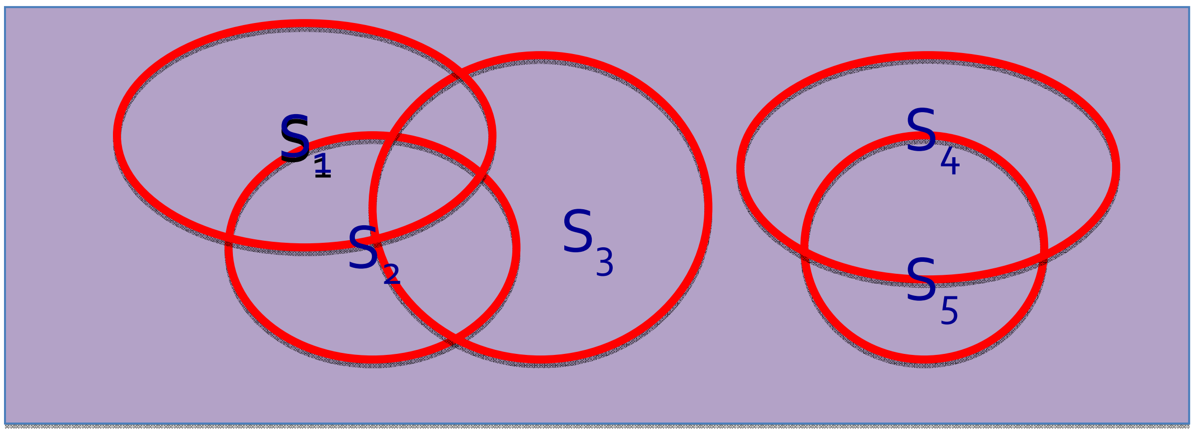
Ranked bandits algorithm.

1. Initialize the k 'bandit algorithms' MAB_1, MAB_2,…,MAB_k
2. For each of the k slots:
   a) select document according to the bandit algorithm.
   b) if already previously chosen, select arbitrary document.
3. Display ordered set of k documents
   a) Assign reward to document if user clicked it and chosen as per the algorithm
   b) Assign penalty otherwise
   c) Update algorithm for the rank

Analysis of the algorithm

Think of this as a maximum k-cover problem.

U



U: User intent expressed as query

S_i: Document d_i

ubmodularity!

Want to find a collection of k sets whose union has maximum cardinality

Which bandit algorithm to use?

Want our algorithm to satisfy the following important criteria

1. Makes no assumptions on distribution of payoffs
2. Allows for exploration strategy
3. Over T rounds, expected payoff of strategies chosen satisfy:

$$\Sigma\, E[f_t(y_t)] \geq \max_y \Sigma\, E[f_t(y)] - R(T)$$

Which bandit algorithm to use?

UCB1 algorithm

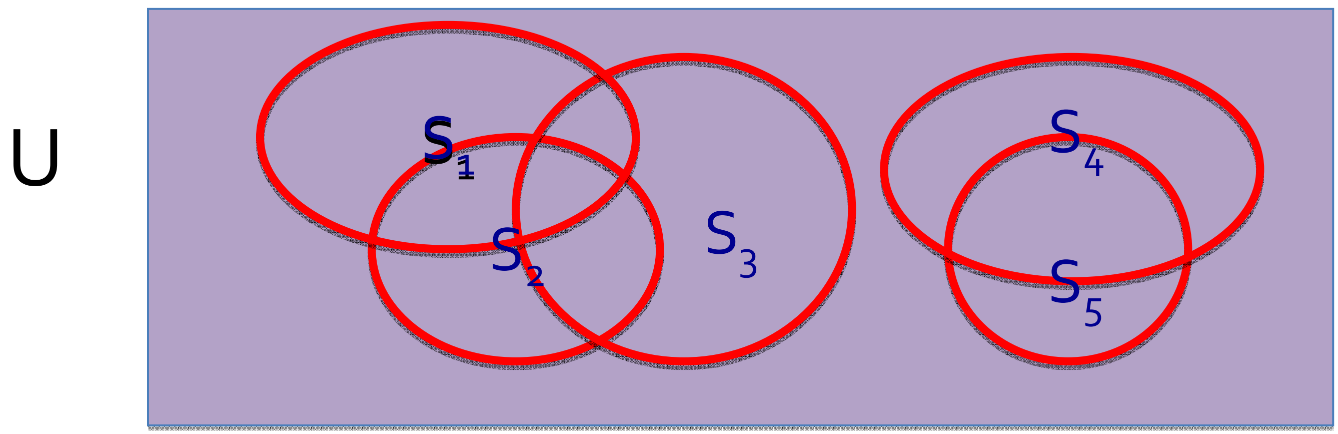
Has the best performance bound of the two candidate choices used

Major weakness: the UCB1 algorithm assumes that the payoffs for the various arms will be i.i.d.

EXP3 algorithm

Exponential-weight multiplicative update algorithm that maintains and updates probabilities of picking arm based on payoffs received

Online maximization of collection of submodular functions (Streeter & Golovin '07)



f_1 f_2 f_3 f_4.... f_n

Want to minimize regret over the choice of each set S_i based on observed payoff given by $f_i(S_i)$

Analysis of the algorithm

Theorem: Ranked Bandits Algorithm achieves a payoff of $(1-1/e)$ OPT – $O(k \sqrt{Tn \log n})$ after T time steps.

Ranked Explore and Commit.

1. Choose some parameters ε, δ and an initial arbitrarily chosen set of k documents
2. For each rank
   a) assign each document to that rank for specified interval and record clicks
   b) increment probability of assigning document that rank if it is chosen by user
   c) choose document with max probability and commit it to the rank
3. Display ordered set of k documents

Analysis of algorithm

Theorem: Ranked explore and commit achieves a payoff of (1-1/e) OPT – εT - $O(nk^3 \log(k/\delta)/\varepsilon)$ after T time steps w.h.p.