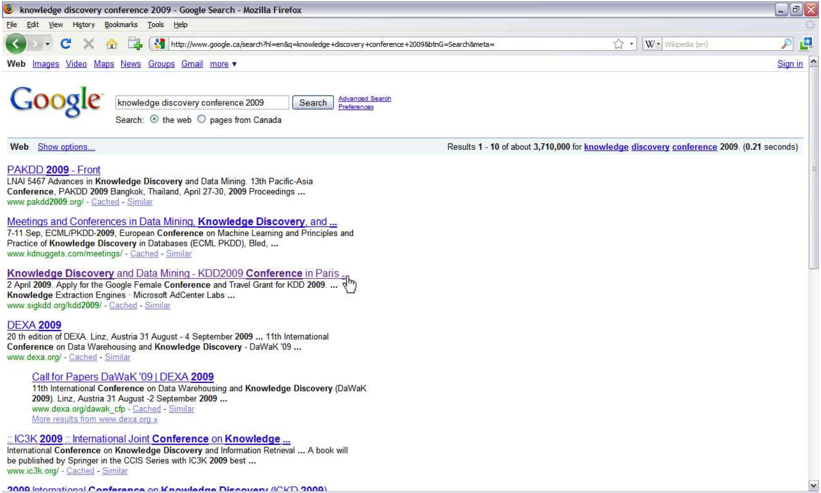
Regret-based Online Ranking for a Growing Digital Library

Erick Delage
Department of Management Sciences
HEC Montréal

June 30, 2009

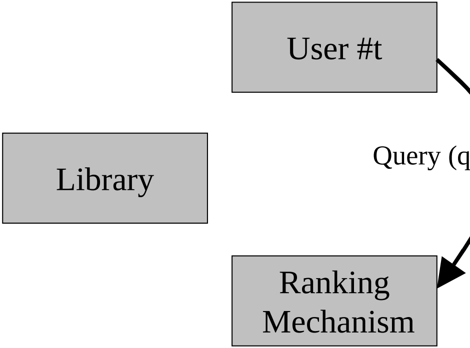
Ranking for Web Search



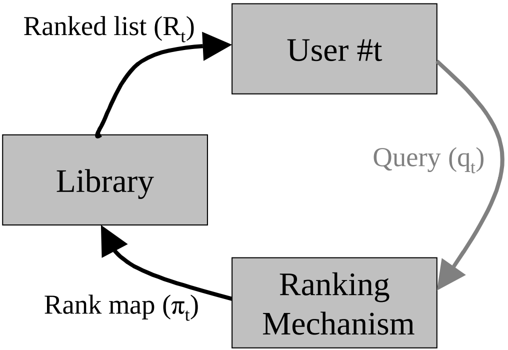
Ranking with Clickthrough Data

- Types of Data:
  - Experts (flexible yet expensive and requires faith in the experts)
  - Clickthrough (abundant yet noisy and subject to manipulation)
- Types of Learning:
  - Offline (uses historical data)
  - Online (uses data as it is provided by users)
- Our model uses clickthrough data assuming that it is truthful

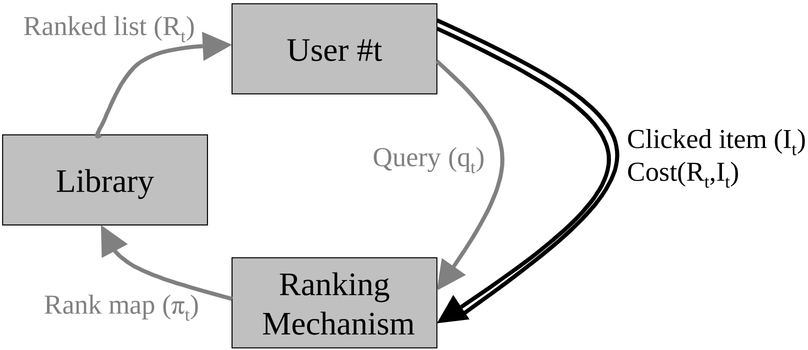
Online Ranking Model I



Online Ranking Model II



Online Ranking Model III



In this Talk

- We will answer the following question:

  "As users are served and the library evolves, does there exist a ranking algorithm which long term performance is comparable to the best ranking chosen with the benefit of hindsight?"

In this Talk

- We will answer the following question:

  > "As users are served and the library evolves, does there exist a ranking algorithm which long term performance is comparable to the best ranking chosen with the benefit of hindsight?"

- Empirical evidence will show that a greedy policy based on the "KL-Rank model" can outperform many popular ranking algorithms.

Outline

1. Introduction
2. The “KL-Rank” Model
3. Reducing Regret in Ranking
4. Experiments with a Citation Database

Outline

Deterministic vs. Non-deterministic Ranking Policies

- Deterministic Ranking
  - Rank items according to a score function

$$score_\theta(I_{\pi(i)}; q) \geq score_\theta(I_{\pi(i+1)}; q) \; \forall \, i \in \{1, 2, ..., |\mathcal{L}|\} \, ,$$

with $score_\theta(I; q) = \sum_i \theta_i \cdot \Phi_i(I, q)$.

Deterministic vs. Non-deterministic Ranking Policies

- Deterministic Ranking
  - Rank items according to a score function

$$score_\theta(I_{\pi(i)}; q) \geq score_\theta(I_{\pi(i+1)}; q) \; \forall \, i \in \{1, 2, ..., |\mathcal{L}|\} \, ,$$

with $score_\theta(I; q) = \sum_i \theta_i \cdot \Phi_i(I, q)$.

- Non-deterministic Ranking
  - Draw a ranking randomly from a distribution

$$\mathbb{P}(\tilde{\pi}(i+1) = j | \tilde{\pi}(1), ..., \tilde{\pi}(i)) \propto \begin{cases} 0 & I_j \in \{I_{\pi(1)}, ..., I_{\pi(i)}\} \\ e^{score_\theta(I_j;q)} & \text{o.w.} \end{cases}$$

Deterministic vs. Non-deterministic Ranking Policies

- Deterministic Ranking
  - Rank items according to a score function

$$score_\theta(I_{\pi(i)}; q) \geq score_\theta(I_{\pi(i+1)}; q) \; \forall \, i \in \{1, 2, ..., |\mathcal{L}|\} \; ,$$

with $score_\theta(I; q) = \sum_i \theta_i \cdot \Phi_i(I, q)$.

- Non-deterministic Ranking
  - Draw a ranking randomly from a distribution

$$\mathbb{P}(\tilde{\pi}(i{+}1) = j | \tilde{\pi}(1), ..., \tilde{\pi}(i)) \propto \begin{cases} 0 & I_j \in \{I_{\pi(1)}, ..., I_{\pi(i)}\} \\ e^{score_\theta(I_j; q)} & \text{o.w.} \end{cases}$$

- Non-deterministic can closely replicate any determ. ranking
- In both cases, generating a ranking costs $O(n \log(n))$

Choosing the Right Cost Model for Ranking

- Common cost models:
  - Fixed cost for not ranking the clicked item first
  - Relative distance of the “clicked” items
  - Neg. normalized discounted cumulated gain (see [Li et al., 2008])
- Finding the best deterministic ranking mechanism is hard (non-continuous & non-convex)

Choosing the Right Cost Model for Ranking

- Common cost models:
  - Fixed cost for not ranking the clicked item first
  - Relative distance of the "clicked" items
  - Neg. normalized discounted cumulated gain (see [Li et al., 2008])
- Finding the best deterministic ranking mechanism is hard (non-continuous & non-convex)
- The problem is continuous for non-deterministic rankings

Choosing the Right Cost Model for Ranking

- Common cost models:
  - Fixed cost for not ranking the clicked item first
  - Relative distance of the "clicked" items
  - Neg. normalized discounted cumulated gain (see [Li et al., 2008])
- Finding the best deterministic ranking mechanism is hard (non-continuous & non-convex)
- The problem is continuous for non-deterministic rankings
- The problem becomes convex when using KL-Cost:

$$\text{KL-Cost}(\tilde{\pi}_\theta, I) = \min_{\nu_{\tilde{\pi}} \in \mathcal{D}(I)} D_{\text{KL}}\left(\nu_{\tilde{\pi}} \| \mu_{\tilde{\pi}_\theta}\right) \quad .$$

(measures the K-L divergence between the ranking distrib. and the set of ranking distrib. that put item I first w.p. one)

Comparing a Heuristic with Best Policy in Hindsight

After $\bar{t}$ users have been served, a good heuristic is to use for user $\bar{t}+1$ a random ranking that is optimal w.r.t. the convex problem:

$$\tilde{\pi}_{\theta_{\bar{t}+1}} \;:\; \theta_{\bar{t}+1} \;=\; \operatorname*{argmin}_{\theta\in\Theta} \sum_{t=1}^{\bar{t}} \text{KL-Cost}(\tilde{\pi}_\theta, I_t)$$

Comparing a Heuristic with Best Policy in Hindsight

After $\bar{t}$ users have been served, a good heuristic is to use for user $\bar{t}+1$ a random ranking that is optimal w.r.t. the convex problem:

$$\tilde{\pi}_{\theta_{\bar{t}+1}} \;:\; \theta_{\bar{t}+1} \;=\; \operatorname*{argmin}_{\theta\in\Theta} \;\sum_{t=1}^{\bar{t}} \text{KL-Cost}(\tilde{\pi}_\theta, I_t)$$

Unfortunately, this approach has no guarantee that in the long run the performance is "close" to optimal (i.e., generates low regret).

$$\mathcal{R}(T) = \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_{\theta_t}, I_t) - \min_{\theta\in\Theta} \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_\theta, I_t) \;=\; ???$$

Outline

Regret in a Fixed Library

- One can choose a feature extracting function $\Phi(\cdot,\cdot)$ and a set of competing ranking policies paramterized by $\theta \in \Theta$.
- If $\max_{q,I} \|\Phi(I,q)\|$ and $\max_{\theta_1,\theta_2\in\Theta} \|\theta_1 - \theta_2\|$ are bounded, then cumulated regret is $O(T^{1/2})$ with “greedy projection”:

$$\theta_{t+1} = \mathcal{P}_\Theta\left(\theta_t - \frac{\alpha}{\sqrt{t}}\nabla_\theta \text{KL-Cost}(\tilde{\pi}_\theta, I_t)\right) .$$

- Average regret is assured to go to zero at the rate of $O(T^{-1/2})$ for any sequence of T user queries. (based on [Zinkevich, 2003])

Regret in a Growing Library

Difficulties:

- As library grows, new feautures are needed to distinguish among the larger set of items
- Regret needs to be measured w.r.t. the features that were available when each user visited:

$$\mathcal{R}(T) = \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_{\theta_t}, I_t) - \min_{\theta} \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_{\mathcal{P}_{\Theta_t}(\theta)}, I_t) \quad .$$

Regret in a Growing Library

Difficulties:

- As library grows, new feautures are needed to distinguish among the larger set of items
- Regret needs to be measured w.r.t. the features that were available when each user visited:

$$\mathcal{R}(T) = \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_{\theta_t}, I_t) - \min_{\theta} \sum_{t=1}^{T} \text{KL-Cost}(\tilde{\pi}_{\mathcal{P}_{\Theta_t}(\theta)}, I_t) \quad .$$

Solution:

- Use "greedy projection with exponential restart".
- Regret cumulated bounded by $O(T^{3/4})$ if, for all $\bar{t} \leq T$,

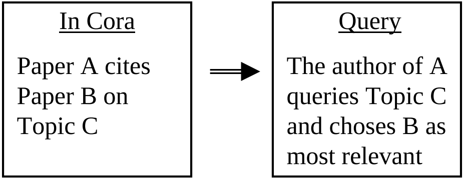
$$\max_{\theta_1, \theta_2 \in \Theta_{\bar{t}}} \|\theta_1 - \theta_2\| \cdot \max_{t < \bar{t}, q, I \in \mathcal{L}_t} \|\Phi_t(I, q)\| = O(\bar{t}^{1/4}) \ .$$

- Avg. regret drops at rate $O(T^{-1/4})$ for any sequence of users.

Outline

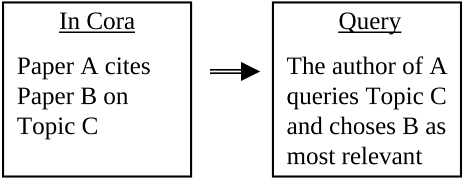
The Cora Dataset

- Cora is an internet portal developed by A. McCallum et al.
- The dataset contains information about 10000 papers from 1913 to 1999: e.g., year of publication, topic, citations.
- We constructed 22000 queries based on each papers citations:



The Cora Dataset

- Cora is an internet portal developed by A. McCallum et al.
- The dataset contains information about 10000 papers from 1913 to 1999: e.g., year of publication, topic, citations.
- We constructed 22000 queries based on each papers citations:



- In implementation, each paper has a dedicated fitness parameter θ_i
- Our theory states that if $\Theta_{\bar{t}} \subseteq \text{Ball}(0, \alpha \bar{t}^{1/4})$, for all $\bar{t} \leq T$, then average regret is $O(T^{-1/4})$.

Competing Algorithms

- *NoRegret KL-Rank*: Rank according to KL-Rank's random policy (guaranteed to lead to no average regret)
- *Greedy KL-Rank*: Rank directly according to score function obtained using KL-Rank

Competing Algorithms

- *NoRegret KL-Rank*: Rank according to KL-Rank's random policy (guaranteed to lead to no average regret)
- *Greedy KL-Rank*: Rank directly according to score function obtained using KL-Rank

- *Online Ranknet*: Online version of RankNet by Burges et al.
- *Category Ranking*: Projection based algorithm proposed by Crammer & Singer
- *HITS & Pagerank*: Graph based ranking algorithms
- *2-class McRank*: Li et al.'s classification method

Statistics obtained in set of 22000 queries

	Avg Rel. Click Dist.		Clicked 1st Rate		Avg NDCG score	
NoRegret KL-Rank	25.7%	(4)	14.3%	(5)	34.4%	(5)
Greedy KL-Rank	19.2%	(1)	17.8%	(1)	39.1%	(1)
Online RankNet	19.8%	(2)	16.1%	(4)	37.4%	(2)
Category Ranking	23.2%	(3)	16.5%	(2)	37.3%	(2)
HITS / PageRank	26.5%	(5)	16.6%	(2)	36.2%	(4)
McRank	26.3%	(5)	14.0%	(5)	34.1%	(5)

- Although Greedy KL-Rank has no performance guarantee (is more opportunistic), it outperforms all other methods.

Conclusion

- Clickthrough data is most natural form of feedback from users and fits well the KL-Rank model
- NoRegret KL-Rank is the first to provide guarantees w.r.t. long term performance (average regret) of policies on arbitrary sequence of users
- Analysis provides guidance for adding features as the library grows while avoiding overfitting
- In practice, Greedy KL-Rank can be less conservative and seems to outperform many popular methods

Conclusion

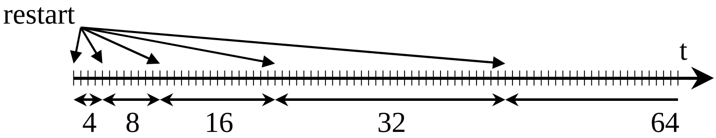
- Clickthrough data is most natural form of feedback from users and fits well the KL-Rank model
- NoRegret KL-Rank is the first to provide guarantees w.r.t. long term performance (average regret) of policies on arbitrary sequence of users
- Analysis provides guidance for adding features as the library grows while avoiding overfitting
- In practice, Greedy KL-Rank can be less conservative and seems to outperform many popular methods
- Future work:
  - Compare methods on other datasets and real-time
  - Study worst case performance guarantees for Greedy KL-Rank
  - Study effects of clickthrough biases and malignant usage

Questions & Comments ...

... Thank you!

Greedy Projection with Exponential Restart Algorithm

Decompose timeline in periods of size that grows exponentially



Let $t = 0$, for cycles $m = 1, 2, \ldots$:

- Keep $\Phi_m(\cdot,\cdot)$ and Θ_m constant over period $t \in [2^{m-1}, 2^m - 1]$
- Initialize $\theta_{t+1} \in \Theta_m$
- For $k = 1, 2, \ldots, 2^{m-1}$:
    1. At time $t = 2^{m-1} + k - 1$, receive query q_t
    2. Generate a random ranking $\tilde{R}_t$ using $\tilde{\pi}_{\theta_t}$
    3. Learn about the preferred item I_t
    4. Update $\theta_{t+1} = \mathcal{P}_{\Theta_m}\left(\theta_t - \frac{\alpha}{\sqrt{k}}\nabla_\theta \text{KL-Cost}(\tilde{\pi}_{\theta_t}, I_t)\right)$