Online Learning with Implicit User Preferences

Thorsten Joachims
Josef Broder, Geri Gay, Laura Granka, Bobby Kleinberg, Madhu Kurup, Filip Radlinski, Pannaga Shivaswamy, Yisong Yue

Department of Computer Science
Cornell University
Today’s Knowledge-Based Systems

• Search Engine, Netflix, Smart Home, Nav Assist, …

• Self-improving Systems
  – Maintenance of knowledge
  – Measure and optimize utility
  → learning

• Sources of Knowledge
  – Expert data collection
  – Tagging
  – Ratings
  – Usage

→ If only I could get 1 (noisy) bit of knowledge out of every user interaction.
Learning for Search Engines

Goal
– Learn improved ranking function

Action of Algorithm
– Present ranking for given query

Feedback from User
– Clicks (also dwell time, scrolling, reformulations, eyetracking)
Overview

Action $a_t$ (e.g. ranking, recommendation)

Goal: Improve $U(a_t)$

Algorithm

User

Feedback

→ Cooperative Online System of Algorithm and User
  - Design choices for Algorithm and Action
  - Models of Users, Feedback, and Utility

→ Outline
  - Learning with Algorithm-Driven Exploration
  - Learning with User-Driven Exploration

User $\rightarrow$ Algorithm
The Most Basic Learning Problem

Distribution $P(u,q)$
of users $u$, queries $q$

Ranking Function 1
$f_1(u,q) \rightarrow r_1$

Which one is better?

Ranking Function 2
$f_2(u,q) \rightarrow r_2$

U(tj,”SVM”,r1)

U(tj,”SVM”,r2)

1. Kernel Machines
   http://svm.first.gmd.de/
2. SVM-Light Support Vector Machine
   http://svmlight.joachims.org/
3. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Service Master Company
   http://www.servicemaster.com/

1. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
2. Service Master Company
   http://www.servicemaster.com/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
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/
Approaches to Implicit Utility Elicitation

• **Approach 1: Absolute Metrics**
  – 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**
  – Do paired comparison tests provide relative preferences between two retrieval functions $f_1$ and $f_2$?
  – For example:
    • $U(f_1) > U(f_2) \Leftrightarrow \text{pairedCompTest}(f_1, f_2) > 0$

[with Filip Radlinski, Mathu Kurup]
How does User Behavior Reflect Retrieval Quality?

User Study in ArXiv.org
- Natural user and query population.
- User in natural context, not lab.
- Live and operational search engine.
- Ground truth by construction

**ORIG** â†’ **SWAP2** â†’ **SWAP4**
- **ORIG**: Hand-tuned fielded
- **SWAP2**: **ORIG** with 2 pairs swapped
- **SWAP4**: **ORIG** with 4 pairs swapped

**ORIG** â†’ **FLAT** â†’ **RAND**
- **ORIG**: Hand-tuned fielded
- **FLAT**: No field weights
- **RAND**: Top 10 of **FLAT** shuffled
How does User Behavior Reflect Retrieval Quality?

User Study in ArXiv.org
– Natural user and query population.
– User in natural context, not lab.
– Live and operational search engine.
– Ground truth by construction

**ORIG** Æ **SWAP2** Æ **SWAP4**
- **ORIG**: Hand-tuned fielded
- **SWAP2**: ORIG with 2 pairs swapped
- **SWAP4**: ORIG with 4 pairs swapped

**ORIG** Æ **FLAT** Æ **RAND**
- **ORIG**: Hand-tuned fielded
- **FLAT**: No field weights
- **RAND**: Top 10 of FLAT shuffled
## Absolute Metrics: Metrics

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Aggregation</th>
<th>Hypothesized Change with Decreased Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandonment Rate</td>
<td>% of queries with no click</td>
<td>N/A</td>
<td>Increase</td>
</tr>
<tr>
<td>Reformulation Rate</td>
<td>% of queries that are followed by reformulation</td>
<td>N/A</td>
<td>Increase</td>
</tr>
<tr>
<td>Queries per Session</td>
<td>Session = no interruption of more than 30 minutes</td>
<td>Mean</td>
<td>Increase</td>
</tr>
<tr>
<td>Clicks per Query</td>
<td>Number of clicks</td>
<td>Mean</td>
<td>Decrease</td>
</tr>
<tr>
<td>Max Reciprocal Rank*</td>
<td>1/rank for highest click</td>
<td>Mean</td>
<td>Decrease</td>
</tr>
<tr>
<td>Mean Reciprocal Rank*</td>
<td>Mean of 1/rank for all clicks</td>
<td>Mean</td>
<td>Decrease</td>
</tr>
<tr>
<td>Time to First Click*</td>
<td>Seconds before first click</td>
<td>Median</td>
<td>Increase</td>
</tr>
<tr>
<td>Time to Last Click*</td>
<td>Seconds before final click</td>
<td>Median</td>
<td>Decrease</td>
</tr>
</tbody>
</table>

(*) only queries with at least one click count
Absolute Metrics: Results

[with Filip Radlinski, Mathu Kurup]
Absolute Metrics: Results

Absolute Metrics: Summary and Conclusions

• None of the absolute metrics reflects expected order.

• Most differences not significant after one month of data.

• Absolute metrics not suitable for ArXiv-sized search engines.

[with Filip Radlinski, Mathu Kurup]
Approaches to Utility Elicitation

• **Approach 1: Absolute Metrics**
  – 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**
  – Do paired comparison tests provide relative preferences between two retrieval functions $f_1$ and $f_2$?
  – For example:
    • $U(f_1) > U(f_2) \iff \text{pairedCompTest}(f_1, f_2) > 0$
Paired Comparisons: What to Measure?

**Interpretation:** \((u=tj,q="svm")\)

\[ f_1(u,q) \rightarrow r_1 \leftarrow (u=tj,q="svm") \]

\[ f_2(u,q) \rightarrow r_2 \]

1. Kernel Machines  
   [http://svm.first.gmd.de/](http://svm.first.gmd.de/)
2. Support Vector Machine  
3. An Introduction to Support Vector Machines  
4. Archives of SUPPORT-VECTOR-MACHINES ...  
   [http://www.jiscmail.ac.uk/lists/SUPPORT...](http://www.jiscmail.ac.uk/lists/SUPPORT...)
5. SVM-Light Support Vector Machine  

**Interpretation:** \( (r_1 \triangleleft r_2) \leftrightarrow \text{clicks}(r_1) > \text{clicks}(r_2) \)

[with Filip Radlinski, Mathu Kurup]
**Paired Comparisons: Balanced Interleaving**

\[ f_1(u,q) \rightarrow r_1 \leftarrow \text{Interleaving}(r_1,r_2) \rightarrow f_2(u,q) \rightarrow r_2 \]

\[(u=tj,q=\text{“svm”})\]

**Interpretation:** \((r_1 \triangleleft r_2) \iff \text{clicks(topk}(r_1)) > \text{clicks(topk}(r_2))\)

**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 \(k_1=k_2 \pm 1\).

[Joachims/01]
Balanced Interleaving: Results

[with Filip Radlinski, Mathu Kurup]
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.
• Same results also for alternative data-preprocessing.

[with Filip Radlinski, Mathu Kurup]
Problem: Learning on Operational System

• Example:
  – 4 retrieval functions: B > G >> 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

[with Yisong Yue, Josef Broder, Bobby Kleinberg]
Regret for the Dueling Bandits Problem

• Given:
  – A finite set $H$ of candidate retrieval functions $f_1 \ldots f_K$
  – A pairwise comparison test $f \triangle f'$ on $H$ with $P(f \triangle f')$

• Regret:
  – $R(A) = \sum_{t=1..T} [P(f^* \triangle f_t) + P(f^* \triangle f'_t) - 1]$
  – $f^*$: best retrieval function (assume single $f^*$ exists)
  – $(f,f')$: retrieval functions tested at time $t$

Example:

<table>
<thead>
<tr>
<th>Time Step</th>
<th>Comparison</th>
<th>Regret</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>$(f_9,f_{12}) \rightarrow f_9$</td>
<td>$P(f^* \triangle f_9) + P(f^* \triangle f_{12}) - 1 = 0.95 + 0.85 - 1 = 0.8$</td>
</tr>
<tr>
<td>$t_2$</td>
<td>$(f_5,f_9) \rightarrow f_5$</td>
<td>$P(f^* \triangle f_5) + P(f^* \triangle f_9) - 1 = 0.91 + 0.82 - 1 = 0.73$</td>
</tr>
<tr>
<td>...</td>
<td>$(f_1,f_3) \rightarrow f_3$</td>
<td>$.5 + .51 - 1 = 0.01$</td>
</tr>
</tbody>
</table>

[with Yisong Yue, Josef Broder, Bobby Kleinberg]
First Thought: Tournament

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

![Diagram with interconnected nodes X1, X2, X3, X4, X5, X6, X7, X8]
Algorithm: Interleaved Filter 2

- **Algorithm**

  InterleavedFilter2(T, W={f₁…fₖ})
  - Pick random f’ from W
  - δ=1/(TK²)
  - WHILE |W|>1
    - FOR b 2 W DO
      » duel(f’,f)
      » update P₉
    - t=t+1
    - c₉=(log(1/δ)/t)⁰.⁵
    - Remove all f from W with P₉ < 0.5-c₉  [WORSE WITH PROB 1-δ]
    - IF there exists f’’ with P₉’’ > 0.5+c₉  [BETTER WITH PROB 1-δ]
      » Remove f’ from W
      » Remove all f from W that are empirically inferior to f’
      » f’=f’’; t=0
  - UNTIL T: duel(f’,f’)

[with Yisong Yue, Josef Broder, Bobby Kleinberg]
Assumptions

• Preference Relation: \( f_i \sim f_j \iff p(f_i \sim f_j) = 0.5 + \varepsilon_{i,j} > 0.5 \)

• Weak Stochastic Transitivity: \( f_i \sim f_j \) and \( f_j \sim f_k \rightarrow f_i \sim f_k \)

\[
f_1 \sim f_2 \sim f_3 \sim f_4 \sim f_5 \sim f_6 \sim \ldots \sim f_K
\]

• Strong Stochastic Transitivity: \( \varepsilon_{i,k} \geq \max\{\varepsilon_{i,j}, \varepsilon_{j,k}\} \)

\[
\varepsilon_{1,4} \geq \varepsilon_{2,4} \geq \varepsilon_{3,4} \geq 0.5 \geq \varepsilon_{5,4} \geq \varepsilon_{6,4} \geq \ldots \geq \varepsilon_{K,4}
\]

• Stochastic Triangle Inequality: \( f_i \sim f_j \sim f_k \rightarrow \varepsilon_{i,k} \leq \varepsilon_{i,j} + \varepsilon_{j,k} \)

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

• \( \varepsilon\)-Winner exists: \( \varepsilon = \max_i\{ p(f_1 \sim f_i) - 0.5 \} = \varepsilon_{1,2} > 0 \)
Assumptions

• Preference Relation: \( f_i \sim f_j \iff P(f_i \sim f_j) = 0.5 + \varepsilon_{i,j} > 0.5 \)

• Weak Stochastic Transitivity: \( f_i \sim f_j \) and \( f_j \sim f_k \) \( \rightarrow \) \( f_i \sim f_k \)

• Strong Stochastic Transitivity: \( \varepsilon_{i,k} \geq \max\{\varepsilon_{i,j}, \varepsilon_{j,k}\} \)

\( \varepsilon_{1,4} \geq \varepsilon_{2,4} \geq \varepsilon_{3,4} \geq 0.5 \geq \varepsilon_{5,4} \geq \varepsilon_{6,4} \geq \ldots \geq \varepsilon_{K,4} \)

• Stochastic Triangle Inequality: \( f_i \sim f_j \sim f_k \) \( \rightarrow \) \( \varepsilon_{i,k} \leq \varepsilon_{i,j} + \varepsilon_{j,k} \)

\( \varepsilon_{1,2} = 0.01 \) and \( \varepsilon_{2,3} = 0.01 \) \( \rightarrow \) \( \varepsilon_{1,3} \leq 0.02 \)

• \( \varepsilon \)-Winner exists: \( \varepsilon = \max_i \{ P(f_1 \sim f_i) - 0.5 \} = \varepsilon_{1,2} > 0 \)

Theorem: IF2 incurs expected average regret bounded by

\[
\frac{1}{T} E(R_T) \leq O \left( \frac{K \log T}{\varepsilon_{1,2} T} \right)
\]
Lower Bound

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

\[ \frac{1}{T} R_T \leq \Omega \left( \frac{K \log T}{\epsilon_1,2T} \right) \]

**Proof:** [Karp/Kleinberg/07][Kleinberg/etal/07]

**Intuition:**
- Magically guess the best bandit, just verify guess
- Worst case: \( 8 \ f_i \hat{A} f_j \): \( P(f_i \hat{A} f_j) = 0.5 + \epsilon \)
- Lemma 2a: Need \( O(1/\epsilon^2 \log T) \) duels to get 1-1/T confidence.
Dueling Bandits: Infinite K

- **Given:**
  - A infinite set $H$ of functions $f_w$ parameterized by $w \in F^{1/2} <^d$
  - A convex value function $v(w)$ describing utility of $f_w$
  - $P(f_w \hat{\Delta} f_{w'}) = \sigma(v(w)-v(w')) = 0.5+\varepsilon(w,w')$

- **Regret:**
  - $R(A) = \sum_{t=1..T} [P(f^* \hat{\Delta} f_t) + P(f^* \hat{\Delta} f'_t) - 1]$
  - $f^*$: best retrieval function in hindsight
  - $(f,f')$: retrieval functions tested at time $t$

- **Approach:**
  - Stochastic estimated gradient method [Zinkevich 03]
  [Flaxman et al. 04]
Dueling Bandits: Infinite K

- **Given:**
  - A infinite set $H$ of functions $f_w$ parameterized by $w \in \mathbb{F}^d$
  - A convex value function $v(w)$ describing utility of $f_w$

- **Regret:**
  - $R(A) = \sum_{t=1}^{T} [P(f^* f_t) + P(f^* f_t')] - 1$
  - $f^*$: best retrieval function in hindsight
  - $(f, f')$: retrieval functions tested at time $t$

- **Approach:**
  - Stochastic estimated gradient method [Zinkevich 03]
    [Flaxman et al. 04]

- **Theorem:** Expected average regret grows as

$$\frac{1}{T} E[R_T] \leq O\left(\frac{\sqrt{WdL}}{T^{1/4}}\right)$$

where $L$=Lipschitz constant for $P$, $d$=dimensionality of parameter space, $W$=bound on norm of $w$
Experiment: Web Search

• Data and Setup
  – Microsoft Web Search dataset with judgments
  – 1000 Queries, 367 Dimensions (Features)
  – Simulate “users” issuing queries

• Results
Overview

Action $a_t$ (e.g. ranking, recommendation)

Goal: Improve $U(a_t)$

Algorithm

User

Feedback

→ Cooperative Online System of Algorithm and User
- Design choices for Algorithm and Action
- Models of Users, Feedback, and Utility

→ Outline
- Learning with Algorithm-Driven Exploration
- Learning with User-Driven Exploration
Who does the exploring?
Who does the exploring?
Who does the exploring?
Strategy for Within-Ranking Feedback

1. Kernel Machines
   http://www.kernel-machines.org/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
4. An Introduction to SVMs
   http://www.support-vector.net/
5. Support Vector Machine and ...
   http://svm.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR...
   http://www.jisc.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.bell-labs.com/SVMsvt.html
8. Royal Holloway SVM
   http://svm.dcs.rhbnc.ac.uk
9. SVM World
   http://www.svmworld.com
10. Fraunhofer FIRST SVM page
    http://svm.first.gmd.de

1. Kernel Machines
   http://www.kernel-machines.org/
2. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
3. Support Vector Machine and ...
   http://svm.bell-labs.com/SVMrefs.html
4. Support Vector Machine
   http://jbolivar.freeservers.com/
5. An Introduction to SVMs
   http://www.support-vector.net/
6. Archives of SUPPORT-VECTOR...
   http://www.jisc.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.bell-labs.com/SVMsvt.html
8. Royal Holloway SVM
   http://svm.dcs.rhbnc.ac.uk
9. SVM World
   http://www.svmworld.com
10. Fraunhofer FIRST SVM page
    http://svm.first.gmd.de
User Study

• Study
  – 36 + 16 subjects (undergrads)
  – Familiar with Google

• Task
  – Answer 10 questions
  – Start with Google search, no restrictions
  – Users unaware of study goal

• Data
  – Queries, clicks, etc.
  – Manual relevance judgments

Who discovered the first modern antibiotic?
Find the homepage of Emeril - the chef who has a TV cooking program.
What actor starred as the main character in the original 'Time Machine' movie?
Find the page displaying the routemap for Greyhound buses.
You are excited to cast your vote in the democratic presidential primary - when can you do so in NY?
Find the homepage of Michael Jordan, the statistician.
Where is the tallest mountain in NY located?
Find the homepage for graduate housing at Carnegie Mellon University.
A friend told you that Mr. Cornell used to live close to campus - between University and Stewart Aves - does anyone live in his house now; if so, who?
Find the homepage of the 1,000 Acres Dude Ranch.

[with Granka, Gay]
Strategy for Within-Ranking Feedback

1. Kernel Machines  
2. Support Vector Machine  
3. SVM-Light Support Vector Machine  
4. An Introduction to SVMs  
5. Support Vector Machine and ...  
   [http://svm.bell-labs.com/SVMrefs.html](http://svm.bell-labs.com/SVMrefs.html)
6. Archives of SUPPORT-VECTOR...  
   [http://www.jisc.ac.uk/lists/SUPPORT...](http://www.jisc.ac.uk/lists/SUPPORT...)
7. Lucent Technologies: SVM demo applet  
8. Royal Holloway SVM  
   [http://svm.dcs.rhbnc.ac.uk](http://svm.dcs.rhbnc.ac.uk)
9. SVM World  
   [http://www.svmworld.com](http://www.svmworld.com)
10. Fraunhofer FIRST SVM page  
    [http://svm.first.gmd.de](http://svm.first.gmd.de)

**Question:** “Click > Skip Above”?  
   e.g. (3>2), (5>2), (5>4)

**Answer:** 78.2 § 5.6 accuracy  
   Inter-Judge: 86.4 accuracy
Strategy for Between-Ranking Feedback

Question: “Click > Top 1 Org”? “Click > Top 2 Org”?

Answer: 85.4 § 8.7 accuracy 84.5 § 6.1 accuracy

1. Kernel Machines
   http://www.kernel-machines.org/
2. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
3. Support Vector Machine and ...
   http://svm.bell-labs.com/SVMrefs.html
4. Service Master Company
   http://www.servicemaster.com/
5. Royal Holloway SVM
   http://svm.dcs.rhbnc.ac.uk
6. Shanghai Velcro Manufacturing
   http://shanghai.velcro.cn/
7. Sport Verein Meppen
   http://fans.svm.de/
8. Society for Vascular Medicine
   http://www.vascmed.org
9. SVM World
   http://www.svmworld.com
10. Fraunhofer FIRST SVM page
    http://svm.first.gmd.de

1. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. Archives of SUPPORT-VECTOR...
   http://www.jisc.ac.uk/lists/SUPPORT...
4. Service Master Company
   http://www.servicemaster.com/
5. Royal Holloway SVM
   http://svm.dcs.rhbnc.ac.uk
6. Shanghai Velcro Manufacturing
   http://shanghai.velcro.cn/
7. Sport Verein Meppen
   http://fans.svm.de/
8. Society for Vascular Medicine
   http://www.vascmed.org
9. SVM World
   http://www.svmworld.com
10. Fraunhofer FIRST SVM page
    http://svm.first.gmd.de
Strategy for Between-Ranking Feedback

Question: “Click > Top 1 Org”? “Click > Top 2 Org”?

Answer: 85.4 § 8.7 accuracy
84.5 § 6.1 accuracy
Does Accuracy Depend on Quality?

- **Controlling Ranking Quality**
  - Normal: Google’s ranking
  - Swapped: Top two documents swap place in every ranking
  - Reversed: Top 10 documents in reversed order in every ranking

→ Subjects permanently assigned to one condition at random

<table>
<thead>
<tr>
<th>Explicit Feedback Data Strategy</th>
<th>Abstracts Phase II “normal”</th>
<th>Abstracts Phase II “swapped”</th>
<th>Abstracts Phase II “reversed”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click &gt; Skip Above</td>
<td>88.0 ± 9.5</td>
<td>79.6 ± 8.9</td>
<td>83.0 ± 6.7</td>
</tr>
<tr>
<td>Click &gt; Top 1 Org</td>
<td>86.4 ± 21.2</td>
<td>77.3 ± 15.1</td>
<td>92.6 ± 16.9</td>
</tr>
<tr>
<td>Click &gt; Top 2 Org</td>
<td>88.9 ± 12.9</td>
<td>80.0 ± 10.1</td>
<td>86.8 ± 12.1</td>
</tr>
</tbody>
</table>
Preference Online Learning Model

- **Model**
  - Unknown user utility function $U(y|x)$: $x$ query, $y$ ranking
  - Optimal $y^* = \text{argmax}_y \{ U(y|x) \}$
  - Loss for prediction $y$: $U(y^*|x) - U(y|x)$
  - LOOP FOREVER
    - Observe context $x$ (e.g. query)
    - Learning algorithm presents $y$
    - Regret = Regret + $[ U(y^*|x) - U(y|x) ]$
    - User returns $y'$ with $U(y'|x) > U(y|x)$

- **Relationship to other online learning models**
  - Expert setting: receive $U(y|x)$ for all $y$
  - Bandit setting: receive $U(y|x)$ only for selected $y$
  - Preference setting: receive single $y'$ with $U(y'|x) > U(y|x)$

[Shivaswamy, Joachims., 11]
Preference Perceptron Algorithm

• Model
  – Linear model of user utility: \( U(y|x) = w^T \hat{A}(x,y) \)

• Algorithm
  • Set \( w_1 = 0 \)
  • FOR \( t = 1 \) TO \( T \) DO
    – Observe \( x_t \)
    – Present \( y_t = \arg\max_y \{ w_t^T \hat{A}(x_t,y) \} \)
    – Obtain feedback \( y'_t \)
    – Update \( w_{t+1} = w_t + \phi(x_t,y'_t) - \phi(x_t,y_t) \)

• This may look similar to a multi-class Perceptron, but
  – Feedback \( y' \) is different (does not get the correct class label)
  – Regret is different (misclassifications vs. utility difference)
    \[
    \frac{1}{T} \sum_{t=1}^{T} [U(y^*_t|x) - U(y_t|x)]
    \]

[Shivaswamy, Joachims., 11]
Preference Perceptron Regret Bound

• Assumptions
  – \( U(y|x) = w^T \phi(x,y) \), but \( w \) is unknown
  – User feedback \( y' \) is \( \alpha \)-informative
    \[
    [U(y'|x) - U(y|x)] \geq \alpha [U(y^*|x) - U(y|x)] - \xi
    \]

• Theorem
  – The average regret of the Preference Perceptron is bounded by
    \[
    \frac{1}{T} \sum_{t=1}^{T} [U(y^*_t|x) - U(y_t|x)] \leq \frac{1}{\alpha T} \sum_{t=1}^{T} \xi_t + \frac{2R||w||}{\alpha \sqrt{T}}
    \]

• Other Algorithms and Results
  – Feedback that is \( \alpha \)-informative only in expectation
  – General convex loss functions of \( U(y^*|x) - U(y|x) \)
  – Regret that scales \( \log(T) \) instead of \( \sqrt{T} \)

[Shivaswamy, Joachims., 11]
Experiment: Web Search

- **Data and Setup**
  - Yahoo! Learning to Rank dataset with judgments \( \text{rel}(d,x) \)
  - Given query \( x \), predict ranking \( y = (d_1, d_2, \ldots) \)
  - Utility:
    - User: \( U(y|x) = \text{DCG@5}(y|x) = \sum_{i=1}^{5} \frac{\text{rel}(d_i, x)}{\log(i + 1)} \)
    - Algorithm: \( U(y|x) = w^T \phi(x, y) = \sum_{i=1}^{5} w^T \psi(d_i, x) \)
Conclusions

Cooperative Online System of Algorithm and User

- Online learning problem with partial observations
- Preferences provide reliable interpretation of user feedback
- Model user as approximately and boundedly rational agent

Settings and Algorithms

- Learning with Algorithm-Driven Exploration
  - Action: pair \((a_t, a_t')\), Utility: order, Feedback: \(P(a_i \neq a_j | U(a_i) > U(a_j))\)
  - Result: \(O(|A| \log(T))\) regret

- Learning with User-Driven Exploration
  - Action: \(a_t = \text{argmax}_a [U_t(a)]\), Utility: linear, Feedback: \(a_t'\) with \(U(a_t') > U(a_t)\)
  - Result: \(O(\|w\| T^{0.5})\) regret

→ Stronger algorithms and results at poster.