Online Learning
DBGD, Pairwise, Historical clicks
Anne Schuth, anne.schuth@uva.nl
External Events can Dwarf Your Changes

Recap (2 weeks ago): control for external events

In this example of an A/B test, you’d be better off with version A
In controlled experiments, both versions are impacted the same way by external events

Oprah calls Kindle “her new favorite thing”

Example: Rich Result Summary

Idea: Users like to know how dishes look like for recipes.
Let's generate nicer captions by including an image.


Recap (2 weeks ago): know what you're measuring
Online Learning to Rank

- When you “go online”, many things beyond your control
- So, control for external events
- Correct for known bias
  - User only click on things they saw
- Know that observations are (very) noisy
  - A click is not always a positive signal
- Especially if you want to (machine) learn from the data you are collecting

In Online L2R

- online != on the web
- online refers to incremental learning; from one instance at a time
Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
What is learning to rank?

A user comes to you with their query.

You look through your index and find 5 trillion matching documents: bear in mind that the average query length is 3 words.

You don't want to put the document the user is looking for on the billionth result page.

So, you need to learn how to rank your documents in the order of relevance to the user.

On the other hand, you can't read minds, so you take a statistical approach and try to learn from data hence the need for (machine) learning.
So, how do you go about doing this?

(slide from earlier this week)

Well, these are two ways:

- Offline Learning (earlier this week): Get lots of pairs \((q,d)\) with relevance information (absolute or relative) and train a regression or classification model.
- Online Learning (today): Learn users' preferences while interacting with them.
Pros and Cons

Offline Learning
- Better studied, so more ML tools.
- Not easily adaptable as things change.
- Often (but not always) the method requires explicit annotations.

Online Learning
- Better for a dynamic environment.
- More sample efficient since the algorithm plays an active role.
- Can't (easily) go back and retrain.
- Personalization becomes possible.
- Preferences come from users with actual information need.
- Applicable in niche settings.
Outline

- Quick Recap
  - Learning to Rank (earlier this week)
  - Online Evaluation (two weeks ago)

- Online Learning to Rank

- Methods
  - Pairwise
    - Balancing Exploration and Exploitation
  - Dueling Bandit Gradient Descent
    - Reusing Historical Interaction Data
    - Optimizing Base Rankers
  - Multileave Gradient Descent
    - Reusing Historical Interaction Data?

- Simulation Framework
relative feedback is much more reliable and thus suitable for online learning

<table>
<thead>
<tr>
<th>Item level</th>
<th>Absolute</th>
<th>Relative</th>
</tr>
</thead>
</table>
|            | Click rate ... | Click-Skip ...
|            | Abandonment ... | A/B testing, interleaving |
Relative / item level

Joachims et al. (2002)
- "Clicked > Skipped Above"
- Preference pairs: #6>#2, #6>#3, #6>#4, #6>#5
- Use Rank SVM to optimize the retrieval function
- Limitation:
  - Confidence of judgments
  - Little implication to user modeling

IR 2013-2014: Evaluation II (Users/Online)
**Interpreting clicks**

(slide from two weeks ago)

Relative feedback is much more reliable and thus suitable for online learning.

<table>
<thead>
<tr>
<th>Item level</th>
<th>Absolute</th>
<th>Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Click rate</td>
<td>Click-Skip</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>SERP level</td>
<td>Abandonment</td>
<td>A/B testing, interleaving</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Baseline: Team draft

(from two weeks ago)

- Goal: Compare two result lists using click data

- Procedure:
  1) Generate interleaved result list (randomize per pair of ranks)
  2) Observe user clicks
  3) Credit clicks to original rankers to infer outcome

\[ o \in \{-1, 0, +1\} \]

Clicks are Biased and Noisy but valuable
(from two weeks ago)

Clicks are biased
- users won’t click on things you didn’t show them
- user are likely to click on things that appear high
- it matters how you present documents
  - snippets, images, colors, font size, grouped with other documents

Clicks are noisy
- they don’t always mean what you hope

Absence of clicks is not always negative
- users might be satisfied due to info in snippet

However: in the long run, clicks do point in the right direction
Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Evaluation

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
Online Learning as Reinforcement Learning

Several methods to learn $w$ online, from clicks

Pairwise

Listwise

Interpret click as positive feedback

Update $w$

$W_{BM25} = 0.9$ would swap $d_1$ and $d_2$

$W = \begin{pmatrix}
  0.5 \\
  0.5 \\
  0.1 \\
  -0.2
\end{pmatrix}$

$w_{BM25} = 0.9$

$\text{s(d|q)}$

$0.58$

$0.36$

$0.24$

$0.26$
Some other things to note

- **Ranker**
  - A “ranker” is a function that, given a query, maps documents to a score on which the documents can be sorted.

- Often, a ranker is a weighted sum of features
  - This is not necessary, our learning methods (only) require that two rankers can be combined
  - See “Optimizing Base Rankers”
  - When the function is fixed, one can refer to the weights themselves as the “ranker”

- **Documents are represented w.r.t. a query by features**
  - Document features (PageRank, #clicks, …)
  - Query-document features (BM25, TFIDF, LM, …)
  - Query-features (query intent, …)
Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Evaluation

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
Cascade model

\[
P(C_i = 1 | E_i = 0) = 0 \\
P(C_i = 1 | E_i = 1) = r_{u_i} \\
P(E_1 = 1) = 1 \\
P(E_i + 1 = 1 | E_i = 0) = 0 \\
P(E_{i+1} = 1 | E_i = 1, C_i = 0) = 1 \\
P(E_{i+1} = 1 | E_i = 1, C_i = 1) = s
\]

several instantiations possible:

- perfect
- navigational
- informational

Outline

- Quick Recap
  - Learning to Rank (earlier this week)
  - Online Evaluation (two weeks ago)

- Online Learning to Rank

- Evaluation

- Methods
  - Pairwise
    - Balancing Exploration and Exploitation
  - Dueling Bandit Gradient Descent
    - Reusing Historical Interaction Data
    - Optimizing Base Rankers
  - Multileave Gradient Descent
    - Reusing Historical Interaction Data?

- Simulation Framework
Pairwise Learning

\[
\hat{w} = \arg\min_{w} \frac{1}{2} \sum_{i=0}^{P} L(w, x_i, y_i) + \frac{1}{2} ||w||^2
\]

where

\[
x = (x_0 - x_0)
\]

\[
L(w, x) = \max(0, 1 - yw^T x)
\]

can be incremental

\[
w_i = w_{i-1} + \eta y_i (x_{i,i} - x_{0,i}) - \eta x_i w_{i-1}
\]

Learn optimal weights

\[
\begin{align*}
0.1 & 0.5 & 0.4 & 0.4 & > & 0.9 & 0.3 & 0.2 & 0.2 & +1 \\
0.001 & 0.9 & 0.9 & 0.9 & < & 0.1 & 0.5 & 0.4 & 0.4 & -1 \\
0.9 & 0.3 & 0.2 & 0.2 & > & 0.001 & 0.9 & 0.9 & 0.9 & +1 \\
0.1 & 0.5 & 0.4 & 0.4 & < & 0.001 & 0.9 & 0.9 & 0.9 & -1 \\
\end{align*}
\]
Pairwise Learning

Algorithm 1: Baseline algorithm for the pairwise setting, based on (Joachims 2002; Sculley 2009; Zhang 2004)

1: Input: $D$, $\eta$, $\lambda$, $w_0$
2: for query $q_i$ ($i = 1..T$) do
3:     $X = \phi(D|q_i)$  // extract features
4:     $S = w_{i,1}^T X$
5:     $L = \text{sortDescendingByScore}(D, S)$
6:     $I = L \{1:10\}$
7:     Display $I$ and observe clicked elements $C$.
8:     Construct all labeled pairs $P = (x_{i,c}, x_{i,b}, y)$ from $I$ and $C$.
9:     for $i$ in $(1..P)$ do
10:        if $y_i w_{i,1}^T (x_{i,c} - x_{i,b}) < 1.0$ and $y_i \neq 0.0$ then
11:           Update $w_i$ as: $w_i = w_{i,1} + \eta y_i (x_{i,c} - x_{i,b}) - \eta \lambda w_{i,1}$
12:     return $w_i$

Pairwise Learning

- Can be purely exploitative
  - We always show the user the best we can
  - And we (still) collect feedback
  - Drawback: position bias [Silverstein et al. 1999]
    - “the higher a document is ranked in the result list presented to the user, the more likely it is to be inspected and clicked.”
    - Thus, many pairs are never observed (as opposed to offline learning)

- Next
  - a way to include exploration in Pairwise Learning
Pairwise Learning

Offline NDCG for the pairwise approach (with 5% confidence intervals)

- Over time
- For the dataset NP2003
- For navigational (a), and informational (b) click models

Outline

- Quick Recap
  - Learning to Rank (earlier this week)
  - Online Evaluation (two weeks ago)
- Online Learning to Rank
- Evaluation
- Methods
  - Pairwise
    - Balancing Exploration and Exploitation
  - Dueling Bandit Gradient Descent
    - Reusing Historical Interaction Data
    - Optimizing Base Rankers
  - Multileave Gradient Descent
    - Reusing Historical Interaction Data?
- Simulation Framework
Dueling Bandit Gradient Descent

- DBGD is a *Listwise* learning to rank method
  - Optimizes the quality of a list of documents
- DBGD uses (confusingly enough) *pairs* of rankers
  - This has the advantage that we can use interleaved comparisons for feedback
  - Hence the “dueling”
Dueling Bandit Gradient Descent

[Yue et al, 2009; Hofmann et al., 2011]
TeamDraft Interleave

Exploitative Ranking
A
B
C
D
E
F

Interleaved Ranking

Explorative Ranking
C
G
D
A
B
E

note: the interleaving method is NOT part of DBGD, it just provides feedback

Dueling Bandit Gradient Descent

Exploitative Ranker

![Graph showing BM25 vs Pagerank for Exploitative Ranker]

Explorative Ranker

![Graph showing BM25 vs Pagerank for Explorative Ranker]

[Yue et al, 2009; Hofmann et al., 2011]
Listwise Learning

Algorithm 3  Baseline Algorithm for the listwise setting

1: Input: \( f(l_1, l_2), x, \delta, w_0 \)
2: for query \( q_i \) (\( i = 1..T \)) do
3:   Sample unit vector \( \omega_i \) uniformly.
4:   \( w'_i \leftarrow w_i + \delta \omega_i \)  // generate exploratory \( w \)
5:   if \( f(l(w_i), l(w'_i)) \) then
6:     \( w_{i+1} \leftarrow w_i + 2\delta \omega_i \)  // update exploitative \( w \)
7: else
8:   \( w_{i+1} \leftarrow w_i \)
9: return \( w_{i+1} \)

Dueling Bandit Gradient Descent

- By nature explorative
  - Unlike pairwise learning, DBGD can not learn from pure exploitation

- Hill climbing
  - Is it a hill we’re climbing?

- Annealing
  - implied

- Queries and clicks are used only once
  - Next, an approach that tries to reuse historical queries and clicks
Pairwise vs Dueling Bandit Gradient Descent

Without tuning exploration, DBGD performs much better.
Outline

- Quick Recap
  - Learning to Rank (earlier this week)
  - Online Evaluation (two weeks ago)

- Online Learning to Rank

- Evaluation

- Methods
  - Pairwise
    - Balancing Exploration and Exploitation
  - Dueling Bandit Gradient Descent
    - Reusing Historical Interaction Data
    - Optimizing Base Rankers
  - Multileave Gradient Descent
    - Reusing Historical Interaction Data?

- Simulation Framework
Dueling Bandit Gradient Descent with Candidate Preselection

Offline performance in NDCG
- computed on held-out test queries after each learning step
- on the TREC NP2003 data set
- for the navigational (b), and informational (c) click models.

Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Evaluation

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
Optimizing Base Rankers

Many “base rankers” (such as BM25, LM) are used out-of-the-box as features for L2R

- But they have parameters
- Optimizing those parameters may have a big impact on performance
- Also these can be learned from clicks

Potential issue

- The ranking functions are no longer linear

Optimizing Base Rankers

Optimization landscape for two parameters of BM25, $k_1$ and $b$

- On the TREC NP2004 data set measured with nDCG
- White crosses indicate where individual runs of the learning algorithm plateaued

Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Evaluation

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
Multileave Gradient Descent

- **Dueling Bandit Gradient Descent**
  - Learns from interleaving feedback
  - Could we learn from multileaving feedback instead?

- **Candidate selection**
  - Random directions / covering area uniformly?
  - How many?

- **Updates**
  - No longer strict winners
  - Pick a winner?
  - Update towards the average of winners?
Multileave Gradient Descent

Offline performance in NDCG
- computed on held-out test queries after each learning step
- on the TREC HP2003 data set
- for the informational click model

Work with Harrie Oosterhuis, to be submitted.
Multileave Gradient Descent
Reusing Historical Interaction Data

- Requires Probabilistic Multileave
  - implemented by some of you
- Collect historical clicks
- Preselect promising candidates
  - How?
  - Project assignment

What would this look like?
Outline

- Quick Recap
  - Learning to Rank (earlier this week)
  - Online Evaluation (two weeks ago)

- Online Learning to Rank

- Evaluation

- Methods
  - Pairwise
    - Balancing Exploration and Exploitation
  - Dueling Bandit Gradient Descent
    - Reusing Historical Interaction Data
    - Optimizing Base Rankers
  - Multileave Gradient Descent
    - Reusing Historical Interaction Data?

- Simulation Framework
Simulation Framework

- Not everyone has access to users
- Some things shouldn’t be tried immediately on real users
  - A decent simulation of users might give you a good understanding of your new fancy algorithm
import sys, random
from lerot import environment, retrieval_system, evaluation, query

user_model = environment.CascadeUserModel(
    [...]  # Details not shown
)
learner = retrieval_system.ListwiseLearningSystem(comparison="comparison.TeamDraft")
evaluation = evaluation.NdcgEval([...])
train = query.load_queries(sys.argv[1], [...])
test = query.load_queries(sys.argv[2], [...])

while True:
    q = train[random.choice(train.keys())]
    l = learner.get_ranked_list(q)
    c = user_model.get_clicks(l, q.get_labeled_positives())
    s = learner.update_solution(c)
pr = evaluation.evaluate_all(s, test)

Learning approaches

- pairwise learning  

- listwise learning
  - dueling bandit gradient descent (Yue and Joachims 2009)
  - candidate preselection (Hofmann et al 2013)
Interleaving methods

- team draft interleave (Radlinski et al 2008)
- document constraints interleave (He et al 2009)
- probabilistic interleave (Hofmann et al 2011)
- optimized interleave (Radlinski and Craswell 2013)
- vertical aware team draft interleave (Chuklin et al 2013)

Multileaving methods
User models

- dependent click model (Guo et al 2009)
- cascade click model (Craswell et al 2008)
- random click model
- federated click model (Chen et al 2012)
Outline

Quick Recap
- Learning to Rank (earlier this week)
- Online Evaluation (two weeks ago)

Online Learning to Rank

Evaluation

Methods
- Pairwise
  - Balancing Exploration and Exploitation
- Dueling Bandit Gradient Descent
  - Reusing Historical Interaction Data
  - Optimizing Base Rankers
- Multileave Gradient Descent
  - Reusing Historical Interaction Data?

Simulation Framework
Questions?

Anne Schuth
anne.schuth@uva.nl
Acknowledgments

- Discussions

- Material
  - Maarten de Rijke, Ron Kohavi, Georg Buscher.

- Support
  - EU FP7 (LiMoSINe), Microsoft Research, Yandex.
References