Learning Ranking Functions with SVMs

CS4780/5780 – Machine Learning
Fall 2014

Thorsten Joachims
Cornell University

T. Joachims, Optimizing Search Engines Using Clickthrough Data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.
http://www.cs.cornell.edu/People/tj/publications/joachims_02c.pdf

Adaptive Search Engines

- Traditional Search Engines
  - One-size-fits-all
  - Hand-tuned retrieval function
- Hypothesis
  - Different users need different retrieval functions
  - Different collections need different retrieval functions
- Machine Learning
  - Learn improved retrieval functions
  - User Feedback as training data

Overview

- How can we get training data for learning improved retrieval functions?
  - Explicit vs. implicit feedback
  - Absolute vs. relative feedback
  - User study with eye-tracking and relevance judgments
- What learning algorithms can use this training data?
  - Ranking Support Vector Machine
  - User study with meta-search engine

Sources of Feedback

- ~~Explicit Feedback~~
  - ~~Overhead for user~~
  - ~~Only few users give feedback => not representative~~
- Implicit Feedback
  - Queries, clicks, time, mousing, scrolling, etc.
  - No Overhead
  - More difficult to interpret

Feedback from Clickthrough Data

Relative Feedback:
Clicks reflect preference between observed links.

Absolute Feedback:
The clicked links are relevant to the query.

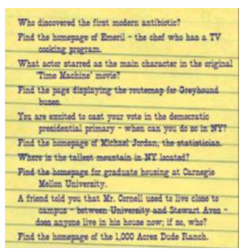
(3 < 2),
(7 < 2),
(7 < 4),
(7 < 5),
(7 < 6)

1. Kernel Machines
   http://svm.first.gmd.de/
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 Support Vector Machines
   http://www.support-vector.net/
5. *Support Vector Machine and Kernel ... References*
   *http://svm.*research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   *http://svm.*research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk

Rel(1),
NotRel(2),
Rel(3),
NotRel(4),
NotRel(5),
NotRel(6),
Rel(7)

User Study: Eye-Tracking and Relevance

- Scenario
  - WWW search
  - Google search engine
  - Subjects were not restricted
  - Answer 10 questions
- Eye-Tracking
  - Record the sequence of eye movements
  - Analyze how users scan the results page of Google
- Relevance Judgments
  - Ask relevance judges to explicitly judge the relevance of all pages encountered
  - Compare implicit feedback from clicks to explicit judgments



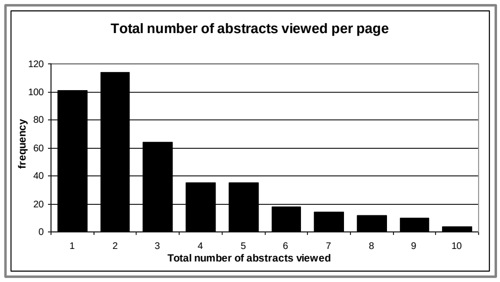
What is Eye-Tracking?

Device to detect and record where and what people look at

- **Fixations**: ~200-300ms; information is acquired
- **Saccades**: extremely rapid movements between fixations
- **Pupil dilation**: size of pupil indicates interest, arousal

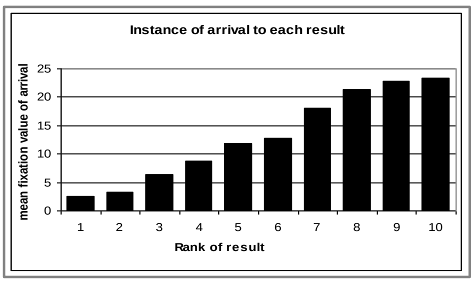
"Scanpath" output depicts pattern of movement throughout screen. Black markers represent fixations.

How Many Links do Users View?



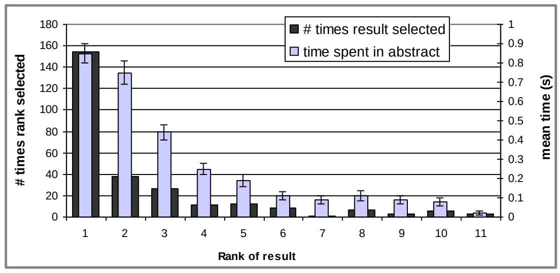
Mean: 3.07 Median/Mode: 2.00

In Which Order are the Results Viewed?



=> Users tend to read the results in order

Looking vs. Clicking



=> Users view links one and two more thoroughly / often
=> Users click most frequently on link one

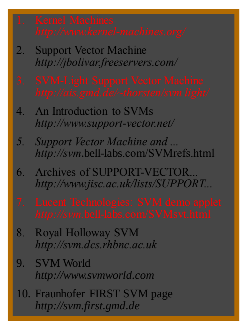
Do Users Look Below the Clicked Link?

Viewed Rank	Clicked Rank 1	2	3	4	5	6
1	90.6%	76.2%	73.9%	60.0%	54.5%	45.5%
2	56.8%	90.5%	82.6%	53.3%	63.6%	54.5%
3	30.2%	47.6%	95.7%	80.0%	81.8%	45.5%
4	17.3%	19.0%	47.8%	93.3%	63.6%	45.5%
5	8.6%	14.3%	21.7%	53.3%	100.0%	72.7%
6	4.3%	4.8%	8.7%	33.3%	18.2%	81.8%

=> Users typically do not look at links below before they click (except maybe the next link)

How do Clicks Relate to Relevance?

- Experiment (Phase II)
  - Additional 16 subjects
  - Manually judged relevance
    - Abstract
    - Page
- Manipulated Rankings
  - **Normal**: Google's ordering
  - **Swapped**: Top Two Swapped
  - **Reversed**: Ranking reversed
- Experiment Setup
  - Same as Phase I
  - Manipulations not detectable



Presentation Bias

~~Hypothesis: Order of presentation influences where users look, but not where they click!~~

“normal”	l_1^-, l_2^-	l_1^+, l_2^-	l_1^-, l_2^+	l_1^+, l_2^+	total
$rel(l_1) > rel(l_2)$	15	19	1	1	36
$rel(l_1) < rel(l_2)$	11	5	2	2	20
$rel(l_1) = rel(l_2)$	19	9	1	0	29
total	45	33	4	3	85
“swapped”	l_1^-, l_2^-	l_1^+, l_2^-	l_1^-, l_2^+	l_1^+, l_2^+	total
$rel(l_1) > rel(l_2)$	11	15	1	1	28
$rel(l_1) < rel(l_2)$	17	10	7	2	36
$rel(l_1) = rel(l_2)$	36	11	3	0	50
total	64	36	11	3	114

Quality-of-Context Bias

~~Hypothesis: Clicking depends only on the link itself, but not on other links.~~

	Rank of clicked link as sorted by relevance judges
Normal + Swapped	**2.67**
Reversed	**3.27**

=> Users click on less relevant links, if they are embedded between irrelevant links.

Are Clicks Absolute Relevance Judgments?

- Clicks depend not only on relevance of a link, but also
  - On the position in which the link was presented
  - The quality of the other links

=> Interpreting Clicks as absolute feedback extremely difficult!

Strategies for Generating Relative Feedback

Strategies
- “Click > Skip Above”
  - **(3>2), (5>2), (5>4)**
- “Last Click > Skip Above”
  - **(5>2), (5>4)**
- “Click > Earlier Click”
  - **(3>1), (5>1), (5>3)**
- “Click > Skip Previous”
  - **(3>2), (5>4)**
- “Click > Skip Next”
  - **(1>2), (3>4), (5>6)**

Comparison with Explicit Feedback

Explicit Feedback Data Strategy	Abstracts Phase I “normal”
Inter-Judge Agreement	89.5
Click > Skip Above	80.8 ± 3.6
Last Click > Skip Above	83.1 ± 3.8
Click > Earlier Click	67.2 ± 12.3
Click > Skip Previous	82.3 ± 7.3
Click > No Click Next	84.1 ± 4.9

=> All but “Click > Earlier Click” appear accurate

Is Relative Feedback Affected by Bias?

Explicit Feedback Data Strategy	Abstracts Phase II “normal”	“swapped”	“reversed”
Click > Skip Above	88.0 ± 9.5	79.6 ± 8.9	83.0 ± 6.7
Last Click > Skip Above	89.7 ± 9.8	77.9 ± 9.9	84.6 ± 6.9
Click > Earlier Click	75.0 ± 25.8	36.8 ± 22.9	28.6 ± 27.5
Click > Skip Previous	88.9 ± 24.1	80.0 ± 18.0	79.5 ± 15.4
Click > No Click Next	75.6 ± 14.5	66.7 ± 13.1	70.0 ± 15.7

⇒Significantly better than random in all conditions, except “Click > Earlier Click”

How Well Do Users Judge Relevance Based on Abstract?

Explicit Feedback	Abstracts	Pages
Data	Phase II	
Strategy	all	all
Inter-Judge Agreement	82.5	86.4
Click > Skip Above	83.1 ± 4.4	78.2 ± 5.6
Last Click > Skip Above	83.8 ± 4.6	80.9 ± 5.1
Click > Earlier Click	46.9 ±13.9	64.3 ±15.4
Click > Skip Previous	81.6 ± 9.5	80.7 ± 9.6
Click > No Click Next	70.4 ± 8.0	67.4 ± 8.2

⇒clicks based on abstracts reflect relevance of the page well

Learning Retrieval Functions from Pairwise Preferences

- Idea: Learn a ranking function, so that number of violated pair-wise training preferences is minimized.
- Form of Ranking Function: sort by

  U(q,d_i) = w_1 * (#of query words in title of d_i)
  + w_2 * (#of query words in anchor)
  + …
  + w_n * (page-rank of d_i)
  = w * Φ(q,d_i)
- Training: Select w so that

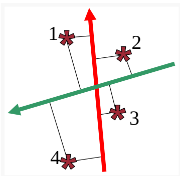
  if user prefers d_i to d_j for query q, then
  U(q, d_i) > U(q, d_j)

Ranking Support Vector Machine

- Find ranking function with low error and large margin

$$
\begin{aligned}
min \quad & \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i,j,k}\xi_{kij} \\
s.t. \quad & \vec{w}\cdot\Phi(q_1,d_i) \geq \vec{w}\cdot\Phi(q_1,d_j) + 1 - \xi_{1ij} \\
& \dots \\
& \vec{w}\cdot\Phi(q_n,d_i) \geq \vec{w}\cdot\Phi(q_n,d_j) + 1 - \xi_{nij}
\end{aligned}
$$

- Properties
  - Convex quadratic program
  - Non-linear functions using Kernels
  - Implemented as part of SVM-light
  - http://svmlight.joachims.org



Experiment

- Meta-Search Engine "Striver"
  - Implemented meta-search engine on top of Google, MSNSearch, Altavista, Hotbot, Excite
  - Retrieve top 100 results from each search engine
  - Re-rank results with learned ranking functions
- Experiment Setup
  - User study on group of ~20 German machine learning researchers and students
    => homogeneous group of users
  - Asked users to use the system like any other search engine
  - Train ranking SVM on 3 weeks of clickthrough data
  - Test on 2 following weeks

Which Ranking Function is Better? Balanced Interleaving

(u=tj, q="svm")

f_1(u,q) → 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) → 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 *http://svm.first.gmd.de/*	1
2.	Support Vector Machine *http://jbolivar.freeservers.com/*	2
3.	SVM-Light Support Vector Machine *http://ais.gmd.de/~thorsten/svm light/*	2
4.	An Introduction to Support Vector Machines *http://www.support-vector.net/*	3
5.	Support Vector Machine and Kernel ... References *http://svm.research.bell-labs.com/SVMrefs.html*	3
6.	Archives of SUPPORT-VECTOR-MACHINES ... *http://www.jiscmail.ac.uk/lists/SUPPORT...*	4
7.	Lucent Technologies: SVM demo applet *http://svm.research.bell-labs.com/SVT/SVMsvt.html*	4

Model of User:
Better retrieval functions is more likely to get more clicks.

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 ± 1.

Interpretation: ($r_1 \succ r_2$) ↔ clicks(topk(r_1)) > clicks(topk(r_2))

[Joachims, 2001] [Radlinski et al., 2008]

Results

Ranking A	Ranking B	A better	B better	Tie	Total
Learned	Google	29	13	27	69
Learned	MSNSearch	18	4	7	29
Learned	Toprank	21	9	11	41

Result:
- Learned > Google
- Learned > MSNSearch
- Learned > Toprank

Toprank: rank by increasing minimum rank over all 5 search engines

Learned Weights

Weight	Feature
0.60	cosine between query and abstract
0.48	ranked in top 10 from Google
0.24	cosine between query and the words in the URL
0.24	doc ranked at rank 1 by exactly one of the 5 engines
...	
0.22	host has the name “citeseer”
...	
0.17	country code of URL is ".de"
0.16	ranked top 1 by HotBot
...	
-0.15	country code of URL is ".fi"
-0.17	length of URL in characters
-0.32	not ranked in top 10 by any of the 5 search engines
-0.38	not ranked top 1 by any of the 5 search engines

Conclusions

- Clickthrough data can provide accurate feedback
  - Clickthrough provides relative instead of absolute judgments
- Ranking SVM can learn effectively from relative preferences
  - Improved retrieval through personalization in meta search
- Current and future work
  - Exploiting query chains
  - Other implicit feedback signals
  - Adapting intranet search for ArXiv.org
  - Recommendation
  - Robustness to “click-spam”
  - Learning and micro-economic theory for interactive learning with preference
  - Further user studies to get better models of user behavior

Feedback across Query Chains