Learning Ranking Functions with SVMs

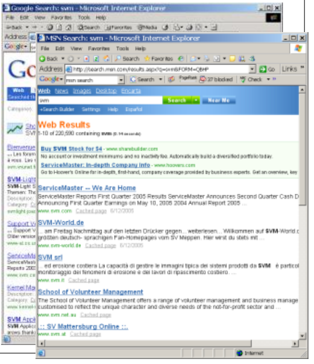
CS4780 – Machine Learning
Fall 2009

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.

Adaptive Search Engines

- **Current 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 Judgements**
  - Ask relevance judges to explicitly judge the relevance of all pages encountered
  - Compare implicit feedback from clicks to explicit judgments

What is Eye-Tracking?

Eye tracking device

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

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

Conclusion: Viewing Behavior

- Users most frequently view two abstracts
- Users typically view results in order from top to bottom
- Users view links one and two more thoroughly and often
- Users click most frequently on link one
- Users typically do not look at links below before they click (except maybe the next link)

=> **Design strategies for interpreting clickthrough data that respect these properties!**

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)**

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

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

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

$$\begin{aligned} rsv(q,d_i) &= w_1 * (\text{\#of query words in title of } d_i) \\ &\quad + w_2 * (\text{\#of query words in anchor}) \\ &\quad + \ldots \\ &\quad + w_n * (\text{page-rank of } d_i) \\ &= w * \Phi(q,d_i) \end{aligned}$$

Training: Select w so that

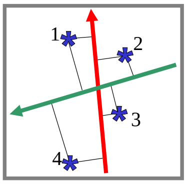
if user prefers d_i to d_i for query q,
then
$rsv(q, d_i) > rsv(q, d_j)$

Ranking Support Vector Machine

- **Find ranking function with low error and large margin**

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

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

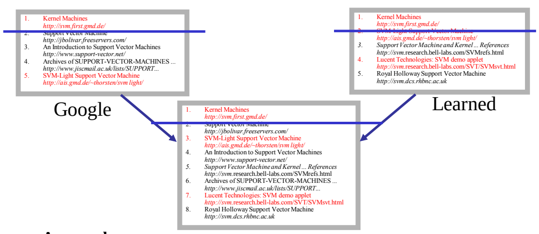


Experiment

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?



- **Approach**
  - Experiment setup generating "unbiased" clicks for fair evaluation.
- **Validity**
  - Clickthrough in combined ranking gives same results as explicit feedback under mild assumptions [Joachims, 2003].

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
  - Adapting intranet search for Cornell Library Web Collection and Physics E-Print ArXiv
  - Implementation of methods in Osmot Search Engine
  - Robustness to "click-spam"
  - Learning theory for interactive learning with preference
  - Further user studies to get more operational model of user behavior
- **Info and Papers**
  - `http://www.joachims.org`