Optimizing Search Engines using Clickthrough Data

Thorsten Joachims

Presented by Botty Dimanov

Overview

1. new algorithm for ranking
2. a way to personalize search engine queries

- Data
- Method
- Experiments

Data Collection

- clickthrough
  - $(q, r, c)\ - q \in$ query, $r \in (\boldsymbol{N} \times$ links$),\ c \equiv clickedLinks \subset domain(r)$
- easy acquisition
- information contained
  - relative relevance

Data Collection

- clickthrough
  - $(q, r, c)\ - q \in$ query, $r \in$ ranking, $c = linksClicked(q, r)$
- easy acquisition
- information contained
  - relative relevance
  - partial relative relevance

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 Methods References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES@JISCMAIL.AC.UK
   http://www.jiscmail.ac.uk/lists/SUPPORT-VECTOR-MACHINES.html
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/
9. Support Vector Machine - The Software
   http://www.support-vector.net/software.html
10. Lagrangian Support Vector Machine Home Page
    http://www.cs.wisc.edu/dmi/lsvm

$$link_3 <_{\mathrm{r}^*} link_2 \qquad link_7 <_{\mathrm{r}^*} link_2$$

$$link_7 <_{\mathrm{r}^*} link_4$$

$$link_7 <_{\mathrm{r}^*} link_5$$

$$link_7 <_{\mathrm{r}^*} link_6$$

$$link_i <_{r^*} link_j$$

for all pairs $1 \leq j < i$, *with* $i \in C$ *and* $j \notin C$.

$$link_3 <_{\mathrm{r}^*} link_2 \qquad \begin{aligned} link_7 &<_{\mathrm{r}^*} link_2 \\ link_7 &<_{\mathrm{r}^*} link_4 \\ link_7 &<_{\mathrm{r}^*} link_5 \\ link_7 &<_{\mathrm{r}^*} link_6 \end{aligned}$$

$$link_i <_{r^*} link_j$$

$$for\ all\ pairs\ 1 \leq j < i,\ with\ i \in C\ \ and\ j \notin C.$$

unsuitable format for machine learning algorithms

Learn Ranking

- How good is a ranking?
  - Kendall's $\tau \equiv \tau(r_a, r_b) = \frac{P - Q}{P + Q} = \frac{1 - 2Q}{\binom{m}{2}}$

$$d_1 <_{r_a} d_2 <_{r_a} d_3 <_{r_a} d_4 <_{r_a} d_5$$

$$d_3 <_{r_b} d_2 <_{r_b} d_1 <_{r_b} d_4 <_{r_b} d_5$$

Learn Ranking

- How good is a ranking?
  - Kendall's $\tau \equiv \tau(r_a, r_b) = \frac{P-Q}{P+Q} = \frac{1-2Q}{\binom{m}{2}}$

$$d_1 <_{r_a} d_2 <_{r_a} d_3 <_{r_a} d_4 <_{r_a} d_5$$

$$d_3 <_{r_b} d_2 <_{r_b} d_1 <_{r_b} d_4 <_{r_b} d_5$$

Learn Ranking

- How good is a ranking?
  - Kendall's $\tau \equiv \tau(r_a, r_b) = \frac{P - Q}{P + Q} = \frac{1 - 2Q}{\binom{m}{2}}$

$$d_1 <_{r_a} d_2 <_{r_a} d_3 <_{r_a} d_4 <_{r_a} d_5$$

$$d_3 <_{r_b} d_2 <_{r_b} d_1 <_{r_b} d_4 <_{r_b} d_5$$

Learn Ranking

- How good is a ranking?
  - Kendall's $\tau \equiv \tau(r_a, r_b) = \frac{P-Q}{P+Q} = \frac{1-2Q}{\binom{m}{2}}$
  - higher τ means higher similarity
  - appropriate measure for IR
  - learn retrieval function

$$\max \tau(f)$$

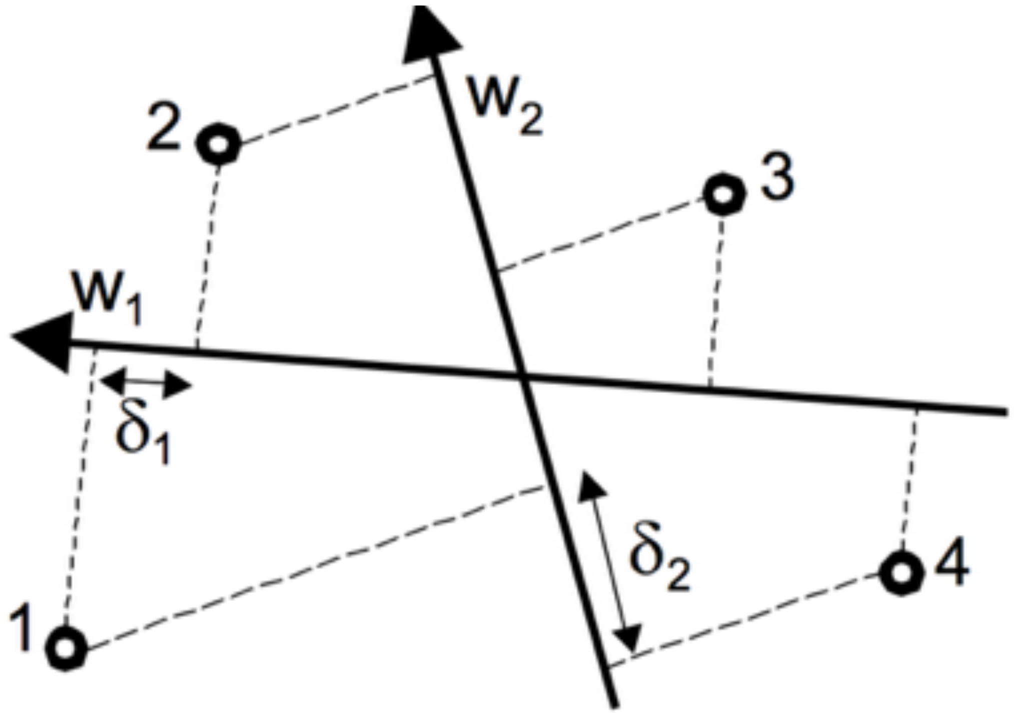
$$\tau(f) = \int \tau\left(r_{f(q)}, r^*\right) d\Pr(q, r^*)$$

$$\tau(f) = \frac{1}{n} \sum_{(q,r^*)} \left(r_{f(q)}, r^*\right)$$

$$Q = |y - f(x)| \sim \tau\left(r_{f(q)}, r^*\right)$$

Retrieval Function

$$(\mathrm{d}_i, \mathrm{d}_j) \in \mathrm{f}_{\vec{w}}(\mathrm{q}) \Longleftrightarrow \vec{w}\Phi(\mathrm{q}, \mathrm{d}_i) > \vec{w}\Phi(\mathrm{q}, \mathrm{d}_j).$$



Optimization problem

- $\max \tau(f) = \frac{1}{n} \sum_{(q,r^*)\in train} \left(r_{f(q)}, r^*\right)$ $\quad = \quad$ $\min \#Q$

Kendall's $\tau \equiv \tau(r_a, r_b) = \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}}$

$$\forall(d_i, d_j) \in r_1^* : \quad \vec{w}\Phi(q_1, d_i) > \vec{w}\Phi(q_1, d_j)$$

$$\cdots$$

$$\forall(d_i, d_j) \in r_n^* : \quad \vec{w}\Phi(q_n, d_i) > \vec{w}\Phi(q_n, d_j)$$

Optimization Problem 1. (Ranking SVM)

$$\textit{minimize:} \qquad V(\vec{w},\vec{\xi}) = \frac{1}{2}\,\vec{w}\cdot\vec{w} + C\sum \xi_{i,j,k} \tag{12}$$

subject to:

$$\begin{aligned}
&\forall (d_i, d_j) \in r_1^* : \vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{i,j,1} \\
&\qquad\qquad\qquad \ldots \\
&\forall (d_i, d_j) \in r_n^* : \vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{i,j,n}
\end{aligned} \tag{13}$$

$$\forall i \forall j \forall k : \xi_{i,j,k} \geq 0 \tag{14}$$

$$\vec{w}\,(\Phi(\mathrm{q}_k, \mathrm{d}_i) - \Phi(\mathrm{q}_k, \mathrm{d}_j)) \geq 1 - \xi_{i,j,k},$$

$$(\mathrm{d}_i, \mathrm{d}_j) \in \mathrm{f}_{\vec{w}^*}(\mathrm{q})$$

$$\iff \vec{w}^* \Phi(\mathrm{q}, \mathrm{d}_i) > \vec{w}^* \Phi(\mathrm{q}, \mathrm{d}_j)$$

$$\iff \sum \alpha^*_{k,l} \Phi(\mathrm{q}_k, \mathrm{d}_l) \Phi(\mathrm{q}, \mathrm{d}_i) > \sum \alpha^*_{k,l} \Phi(\mathrm{q}_k, \mathrm{d}_l) \Phi(\mathrm{q}, \mathrm{d}_j)$$

$$rsv(\mathrm{q}, \mathrm{d}_i) = \vec{w}^* \Phi(\mathrm{q}, \mathrm{d}_i) = \sum \alpha^*_{k,l} \Phi(\mathrm{q}_k, \mathrm{d}_l) \Phi(\mathrm{q}, \mathrm{d}_j)$$

Experiments

- Offline – SVM learn a retrieval function f . $\max$ Kendel's τ
- Online - $\max$ Kendel's τ improves retrieval quality

- "Striver" - combination method

Ranking A:

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.....com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http://svm.....com/SVT/SVMsvt.html
5. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/
6. Support Vector Machine - The Software
   http://www.support–vector.net/software.html
7. Support Vector Machine - Tutorial
   http://www.support–vector.net/tutorial.html
8. Support Vector Machine
   http://jbolivar.freeservers.com/

Ranking B:

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/
6. Support Vector Machine - The Software
   http://www.support–vector.net/software.html
7. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm
8. A Support ... - Bennett, Blue (ResearchIndex)
   http://citeseer.../bennett97support.html

Combined Results:

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 Methods References
   http://svm.research.bell − labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES@JISCMAIL.AC.UK
   http://www.jiscmail.ac.uk/lists/SUPPORT–VECTOR–MACHINES.html
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/
9. Support Vector Machine - The Software
   http://www.support − vector.net/software.html
10. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm

rank_X: 100 minus rank in $X \in$ {Google, MSN-Search, Altavista, Hotbot, Excite} divided by 100 (minimum 0)

top1_X: ranked #1 in $X \in$ {Google, MSNSearch, Altavista, Hotbot, Excite} (binary $\{0, 1\}$)

top10_X: ranked in top 10 in $X \in$ {Google, MSN-Search, Altavista, Hotbot, Excite} (binary $\{0, 1\}$)

top50_X: ranked in top 50 in $X \in$ {Google, MSN-Search, Altavista, Hotbot, Excite} (binary $\{0, 1\}$)

top1count_X: ranked #1 in X of the 5 search engines

top10count_X: ranked in top 10 in X of the 5 search engines

top50count_X: ranked in top 50 in X of the 5 search engines

2. Query/Content Match (3 features total):

query_url_cosine: cosine between URL-words and query (range $[0, 1]$)

query_abstract_cosine: cosine between title-words and query (range $[0, 1]$)

domain_name_in_query: query contains domain-name from URL (binary $\{0, 1\}$)

3. Popularity-Attributes ($\sim$ 20.000 features total):

url_length: length of URL in characters divided by 30

country_X: country code X of URL (binary attribute $\{0, 1\}$ for each country code)

domain_X: domain X of URL (binary attribute $\{0, 1\}$ for each domain name)

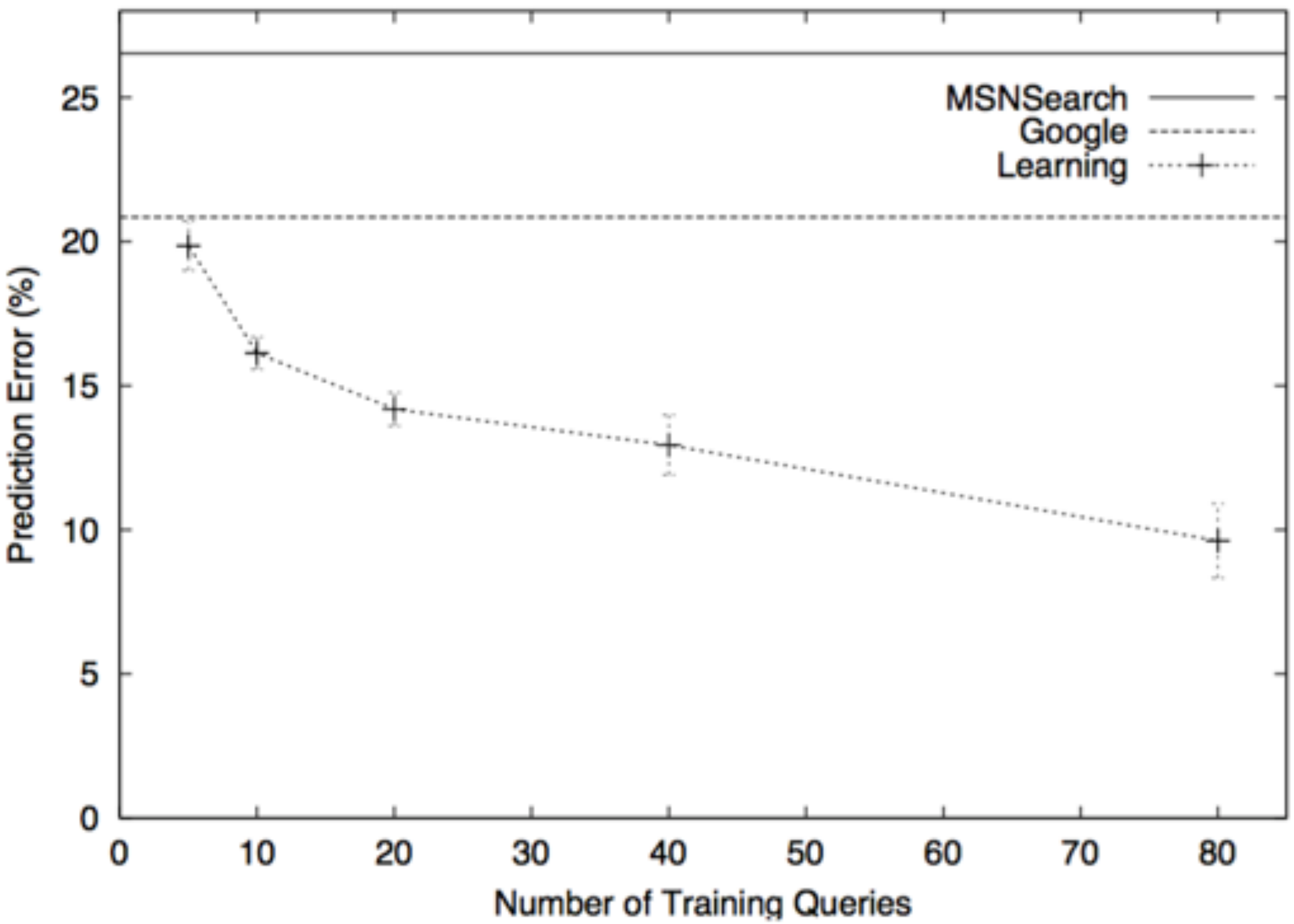
abstract_contains_home: word "home" appears in URL or title (binary attribute $\{0, 1\}$)

url_contains_tilde: URL contains "~" (binary attribute $\{0, 1\}$)

url_X: URL X as an atom (binary attribute $\{0, 1\}$)

Offline

- Combination Google + MSN Search
- 112 queries
- ranked $d_i <_r$ random d_j


MSNSearch
Google
Learning
Prediction Error (%)
Number of Training Queries
0
5
10
15
20
25
30
40
50
60
70
80

Online

- training - 216 queries
- evaluation – Combination (Learned + random SE)

Comparison	more clicks on learned	less clicks on learned	tie (with clicks)	no clicks	total
Learned vs. Google	29	13	27	19	88
Learned vs. MSNSearch	18	4	7	11	40
Learned vs. Toprank	21	9	11	11	52

weight	feature
0.60	query_abstract_cosine
0.48	top10_google
0.24	query_url_cosine
0.24	top1count_1
0.24	top10_msnsearch
0.22	host_citeseer
0.21	domain_nec
0.19	top10count_3
0.17	top1_google
0.17	country_de
...	
0.16	abstract_contains_home
0.16	top1_hotbot
...	
0.14	domain_name_in_query
...	
-0.13	domain_tu-bs
-0.15	country_fi
-0.16	top50count_4
-0.17	url_length
-0.32	top10count_0
-0.38	top1count_0

Table 3: Features with largest and smallest weights as learned from the training data in the online experiment.

Conclusion

- Ranking SVM can successfully learn an improved retrieval function
- function automatically adapts
- Good results – taking the best of all search engines