

# Learning Ranking Functions with Implicit Feedback

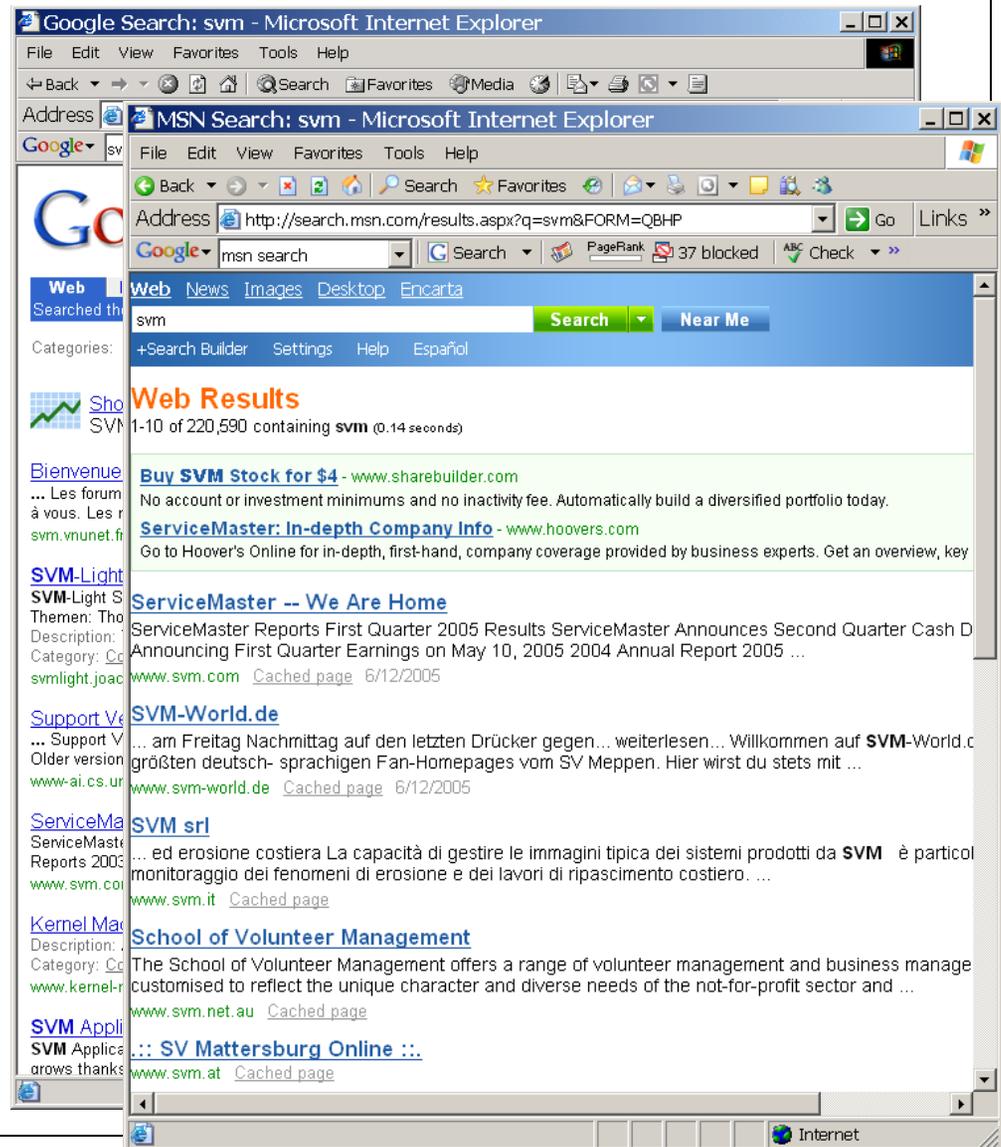
**CS4780 – Machine Learning  
Fall 2011**

**Pannaga Shivaswamy  
Cornell University**

These slides are built on an earlier set of slides by Prof. Joachims.

# 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

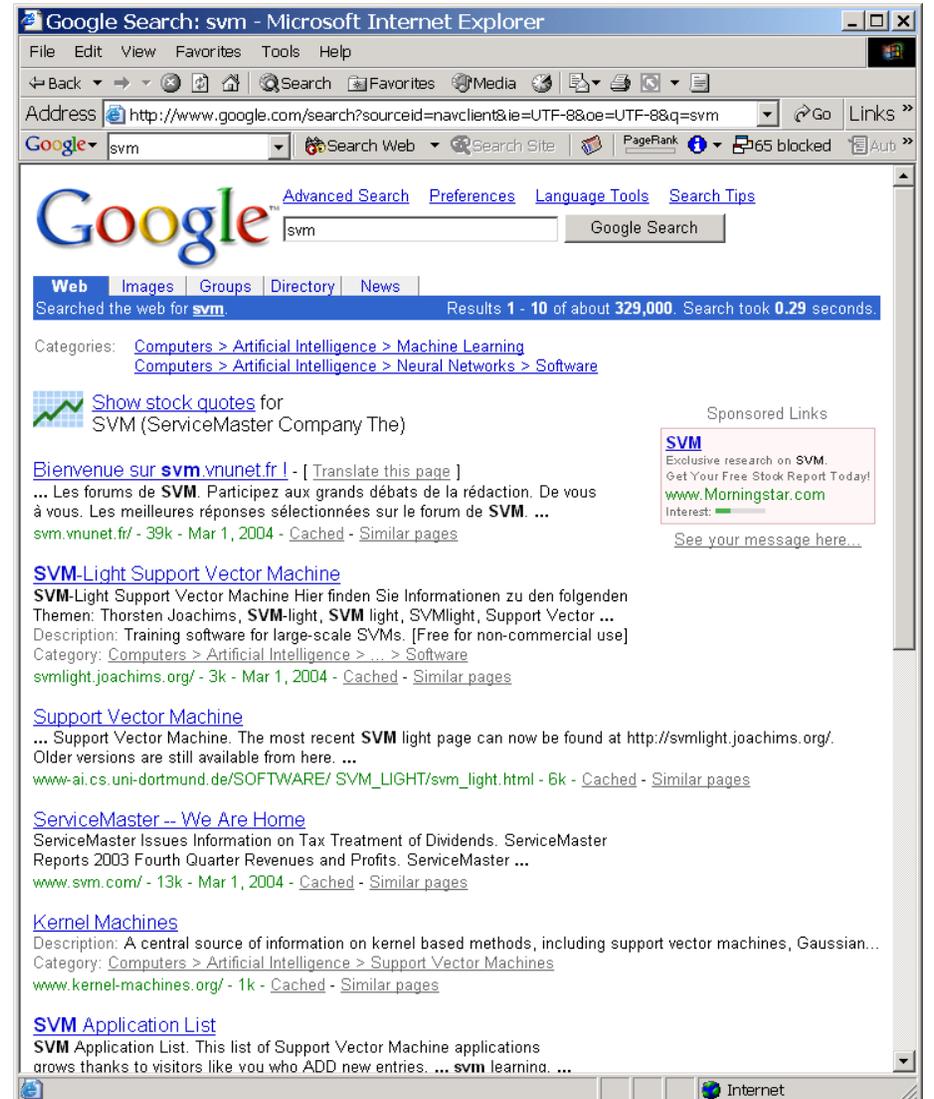


# Overview

- **How can we get training data for learning improved retrieval functions?**
  - Explicit vs. implicit feedback
  - Absolute vs. relative feedback
- **Learning to rank algorithms:**
  - Ranking Support Vector Machine
  - Online learning to rank algorithm with interactive feedback

# 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



# Types of Feedback

- **Absolute Feedback**
  - Feedback about relevance of document on absolute scale
  - Examples
    - Document  $d_i$  is relevant to query  $q$
    - Document  $d_j$  is not relevant to query  $q$
    - Document  $d_l$  has relevance 0.73 with respect to query  $q$
- **Relative Feedback**
  - Feedback reflects preference between documents
  - Examples
    - Document  $d_i$  is more relevant to query  $q$  than document  $d_j$
    - Document  $d_i$  is the least relevant to query  $q$  among  $\{d_i, d_j, d_l, d_m\}$

# Feedback from Clickthrough Data

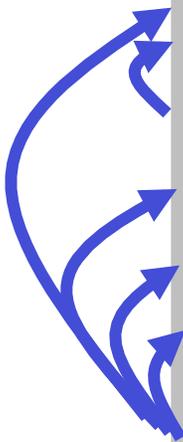
## Relative Feedback:

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.freesevers.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)

# 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.freesevers.com/>
3. SVM-Light Support Vector Machine  
[http://ais.gmd.de/~thorsten/svm\\_light/](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>

# Learning Retrieval Functions with SVMs

**Form of retrieval function:** query  $q$ , documents  $d_1, \dots, d_k$   
sort by  $score(q, d_i) = w_1 * (\# \text{of query words in title of } d_i)$   
 $+ w_2 * (\# \text{of query words in anchor})$   
 $+ \dots$   
 $+ w_n * (\text{page-rank of } d_i)$   
 $= w * \Phi(q, d_i)$

## Absolute Feedback:

- Binary Classification SVM
  - Rel vs. NotRel
- Learn score so that
  - $score(q, d_i) > 0$ , if Rel
  - $score(q, d_i) < 0$ , if NotRel

## Relative Feedback:

- Ranking SVM
- Learn score so that
  - $score(q, d_i) > score(q, d_j)$   
 $\Leftrightarrow$   
 $d_i$  more relevant than  $d_j$

# 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

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

**Training:** Select  $w$  so that

if user prefers  $d_i$  to  $d_j$  for query  $q$ ,  
then

$$\text{rsv}(q, d_i) > \text{rsv}(q, d_j)$$

# Ranking Support Vector Machine

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

$$\min \quad \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum_{i,j,k} \xi_{kij}$$

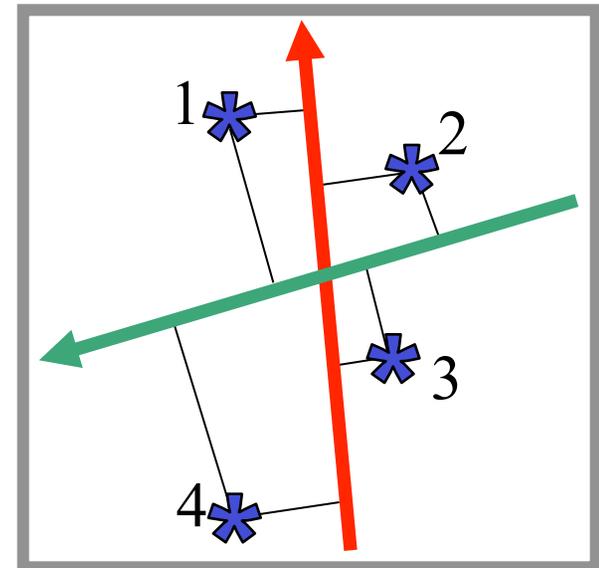
$$s.t. \quad \vec{w} \cdot \Phi(q_1, d_i) \geq \vec{w} \cdot \Phi(q_1, d_j) + 1 - \xi_{1ij}$$

...

$$\vec{w} \cdot \Phi(q_n, d_i) \geq \vec{w} \cdot \Phi(q_n, d_j) + 1 - \xi_{nij}$$

- **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?

- 
1. Kernel Machines  
<http://svm.first.gmd.de/>
  2. Support vector machine  
<http://jbolivar.freesevers.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/](http://ais.gmd.de/~thorsten/svm_light/)

Google

1. Kernel Machines  
<http://svm.first.gmd.de/>
2. SVM-Light Support Vector Machine  
[http://ais.gmd.de/~thorsten/svm\\_light/](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>

Learned

1. Kernel Machines  
<http://svm.first.gmd.de/>
2. Support vector machine  
<http://jbolivar.freesevers.com/>
3. SVM-Light Support Vector Machine  
[http://ais.gmd.de/~thorsten/svm\\_light/](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>

- **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

<b>Ranking A</b>	<b>Ranking B</b>	<b>A better</b>	<b>B better</b>	<b>Tie</b>	<b>Total</b>
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

# Learned Weights

- | <b>Weight</b> | <b>Feature</b>                                       |
|---------------|--|
| • 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      |

# Online learning with interactive feedback

**Idea:** Learn a ranking function online based on interaction from the user

**Present a ranking**  $\rightarrow$  **Observe user feedback**  $\rightarrow$  **Update**  
 $y$   $\bar{y}$

**Goal:** Present good rankings to a user

## **Advantages of online learning to rank:**

- ✓ Naturally suited to how a user interacts
- ✓ No need of an expensive offline training data
- ✓ Theoretical guarantees
- ✓ Performs well compared to offline learning

# Listwise feedback

(1,3,7,2,4,5,6,8) > (1,2,3,4,5,6,7,8)

- 
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/](http://ais.gmd.de/~thorsten/svm%20light/)
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>

# Learning Model

**Utility of a ranking to a query as a linear model:**

$$\mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t)$$

**Optimal ranking for a query:**

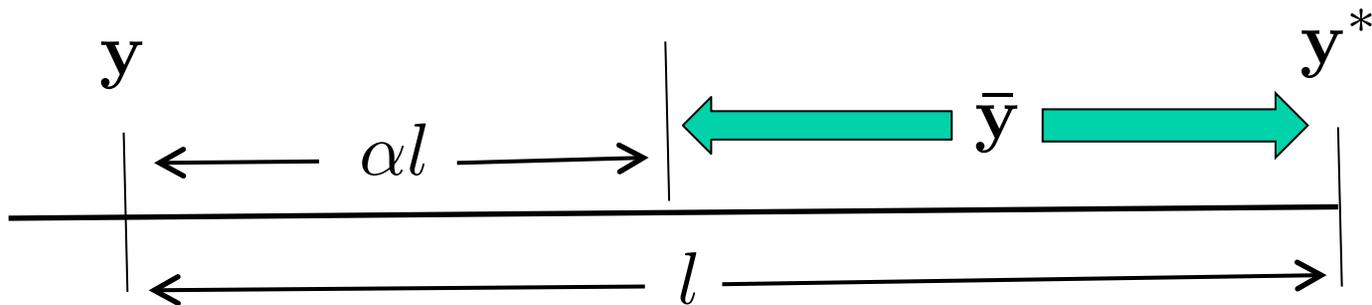
$$\mathbf{y}_t^* \leftarrow \arg \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y})$$

**$\alpha$  informative feedback:**

$$\begin{aligned} & \mathbf{w}^\top \Phi(\mathbf{q}_t, \bar{\mathbf{y}}_t) - \mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t) \\ & \geq \alpha (\mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t)) \end{aligned}$$

## User Feedback

$$\begin{aligned} & \mathbf{w}^\top \Phi(\mathbf{q}_t, \bar{\mathbf{y}}_t) - \mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t) \\ & \geq \alpha (\mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \mathbf{w}^\top \Phi(\mathbf{q}_t, \mathbf{y}_t)) \end{aligned}$$



# Interactive Perceptron Algorithm

Initialize  $\mathbf{w}_1 \leftarrow 0$

**for**  $t = 1$  **to**  $T$  **do**

    Observe  $\mathbf{q}_t$

    Present  $\mathbf{y}_t \leftarrow \arg \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}_t^\top \Phi(\mathbf{q}_t, \mathbf{y})$

    Receive feedback  $\bar{\mathbf{y}}_t$

    Update  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Phi(\mathbf{q}_t, \bar{\mathbf{y}}_t) - \Phi(\mathbf{q}_t, \mathbf{y}_t)$

**end for**

# Theoretical Guarantee

**Notion of regret:**

$$\frac{1}{T} \sum_{t=1}^T \mathbf{w}^\top (\Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \Phi(\mathbf{q}_t, \mathbf{y}_t))$$

**Conditions:**

$\alpha$  informative feedback

boundedness of features  $\|\Phi(\mathbf{q}, \mathbf{y})\|^2 \leq R^2$

**Guarantee:**

$$\text{Regret} \leq \frac{2R\|\mathbf{w}\|}{\alpha\sqrt{T}}$$

## Proof

$$\begin{aligned} & \mathbf{w}_{T+1}^\top \mathbf{w}_{T+1} \\ &= \mathbf{w}_T^\top \mathbf{w}_T + 2\mathbf{w}_T^\top (\Phi(\mathbf{q}_T, \bar{\mathbf{y}}_T) - \Phi(\mathbf{q}_T, \mathbf{y}_T)) \\ & \quad + (\Phi(\mathbf{q}_T, \bar{\mathbf{y}}_T) - \Phi(\mathbf{q}_T, \mathbf{y}_T))^\top (\Phi(\mathbf{q}_T, \bar{\mathbf{y}}_T) - \Phi(\mathbf{q}_T, \mathbf{y}_T)) \\ & \leq \mathbf{w}_T^\top \mathbf{w}_T + 4R^2 \leq \mathbf{w}_{T-1}^\top \mathbf{w}_{T-1} + 2.4R^2 \\ & \leq \mathbf{w}_1^\top \mathbf{w}_1 + 4R^2 T \leq 4R^2 T \end{aligned} \leq 0$$

$$\|\mathbf{w}_{T+1}\| \leq 2R\sqrt{T}$$

Recall :  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Phi(\mathbf{q}_t, \bar{\mathbf{y}}_t) - \Phi(\mathbf{q}_t, \mathbf{y}_t)$

$$\|\Phi(\mathbf{q}, \mathbf{y})\|^2 \leq R^2 \quad \mathbf{w}_1 = 0$$

From the assumption on feedback

$$\begin{aligned}\mathbf{w}_{T+1}^\top \mathbf{w} &= \sum_{t=1}^T \mathbf{w}^\top (\Phi(\mathbf{q}_t, \bar{\mathbf{y}}_t) - \Phi(\mathbf{q}_t, \mathbf{y}_t)) \\ &\geq \alpha \sum_{t=1}^T \mathbf{w}^\top (\Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \Phi(\mathbf{q}_t, \mathbf{y}_t))\end{aligned}$$

Therefore,

$$\begin{aligned}\alpha \sum_{t=1}^T \mathbf{w}^\top (\Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \Phi(\mathbf{q}_t, \mathbf{y}_t)) &\leq \mathbf{w}^\top \mathbf{w}_{T+1} \\ &\leq \|\mathbf{w}\| \|\mathbf{w}_{T+1}\| \leq 2R\sqrt{T} \|\mathbf{w}\|\end{aligned}$$

Recall :

$$\mathbf{w}^\top \mathbf{w}_{T+1} \leq \|\mathbf{w}\| \|\mathbf{w}_{T+1}\|$$

$$\|\mathbf{w}_{T+1}\| \leq 2R\sqrt{T}$$

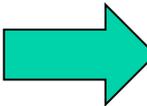
Rearranging

$$\frac{1}{T} \sum_{t=1}^T \mathbf{w}^\top (\Phi(\mathbf{q}_t, \mathbf{y}_t^*) - \Phi(\mathbf{q}_t, \mathbf{y}_t)) \leq \frac{2R\|\mathbf{w}\|}{\alpha\sqrt{T}}$$

As  $T \rightarrow \infty$  average regret approaches 0

## Experiment

### Dataset:

Yahoo! Learning to rank dataset 

### Utility of a ranking:

$$\mathbf{w}^\top \Phi(\mathbf{q}, \mathbf{y}) = \sum_{i=1}^5 \frac{\mathbf{w}^\top \phi(\mathbf{q}, d_{y_i})}{\log(1 + i)}$$

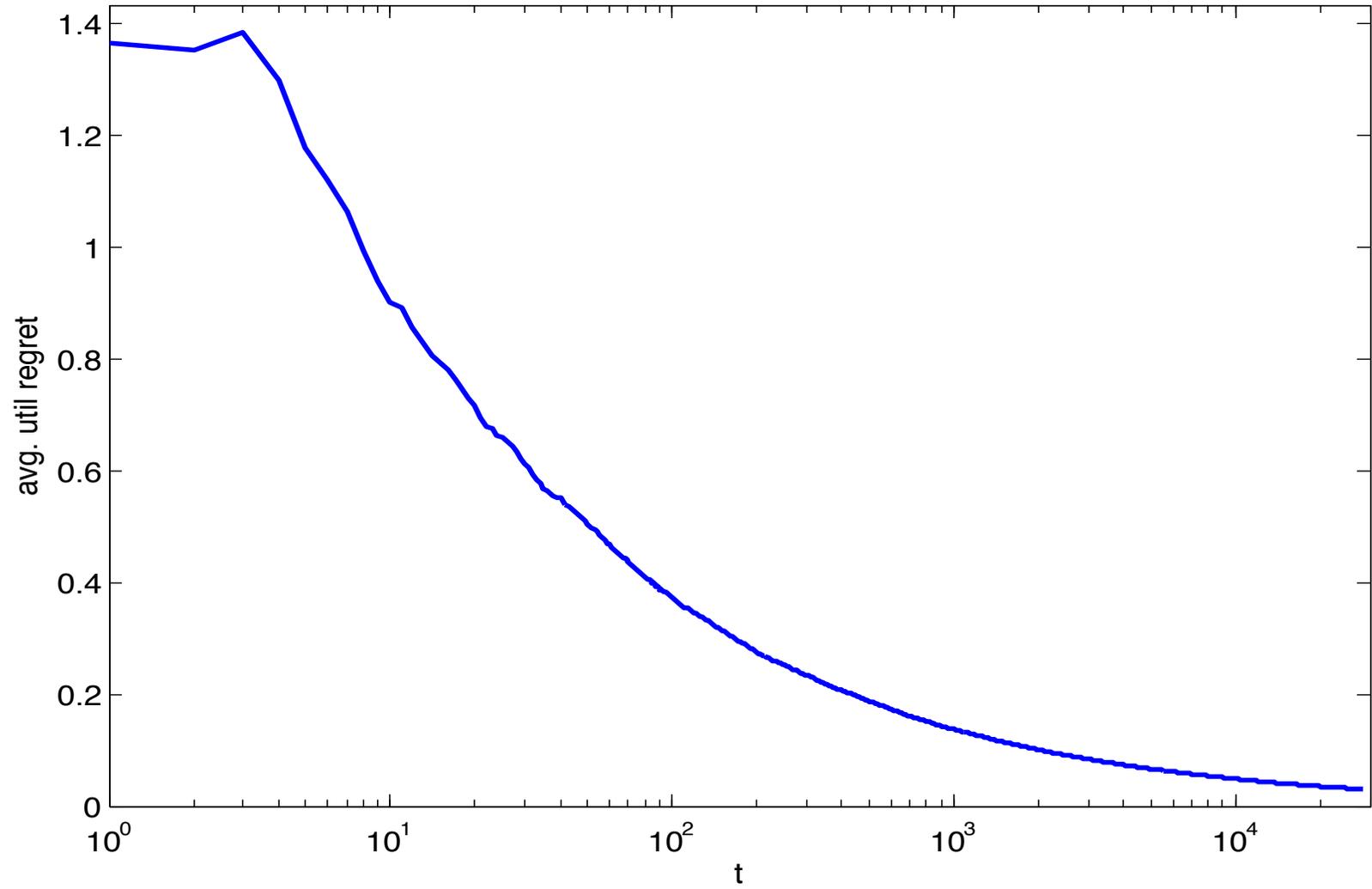
$\mathbf{w}$   best least squares fit on the data

### Baseline:

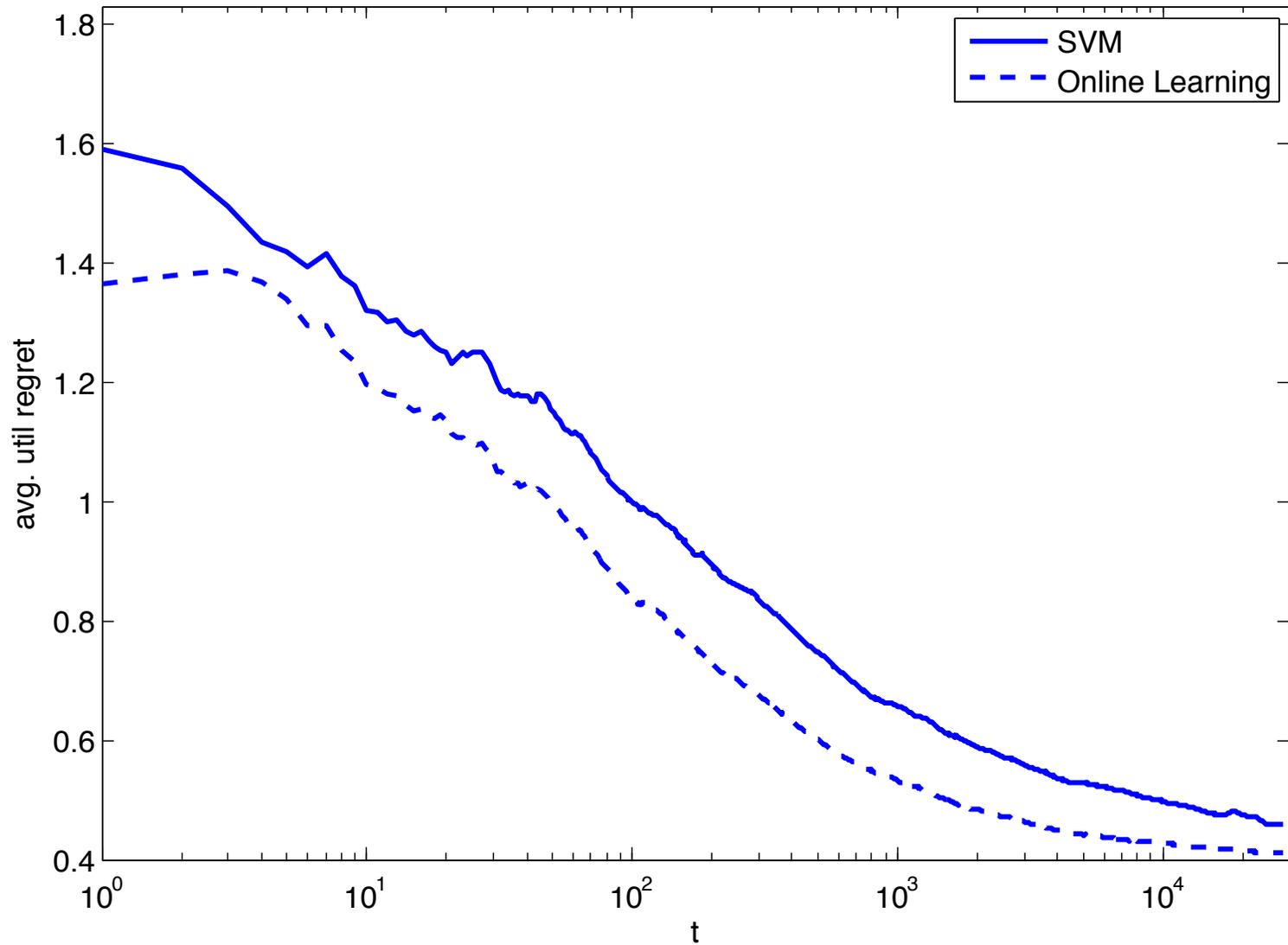
Pairwise SVM trained repeatedly

(query,URL)	relevance
(q_1,d_1)	0
(q_1,d_2)	3
(q_1,d_3)	4
(q_2,d_1)	1
(q_2,d_2)	1
(q_2,d_3)	0
(q_2,d_4)	2

# Clean Feedback



# Noisy Feedback



# 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
- **Online learning is very natural in this setting**
  - Performs better than repeatedly training an SVM
- **Current and future work**
  - Applying the online learning model to a real-world setup
  - Diversity in search results via online learning
  - Better theoretical guarantees for particular families of loss functions
  - Incorporating similarity information across users