LEARNING TO RANK FROM CLICKTHROUGH DATA

Qiuyan Xu

Overview

- Logging
- Feedback
  - Rating
  - Preferences
  - Ranking
- Learning to Rank
  - Point-wise
  - Pair-wise
  - List-wise
- Performance

Motivation

- Machine learning approaches are applied in learning to rank
- They achieve better performances as conventional methods
- However, a large quantity of training data is required
- How to win the training data?
- How to learn with those data?

We May Call Some Experts to Judge

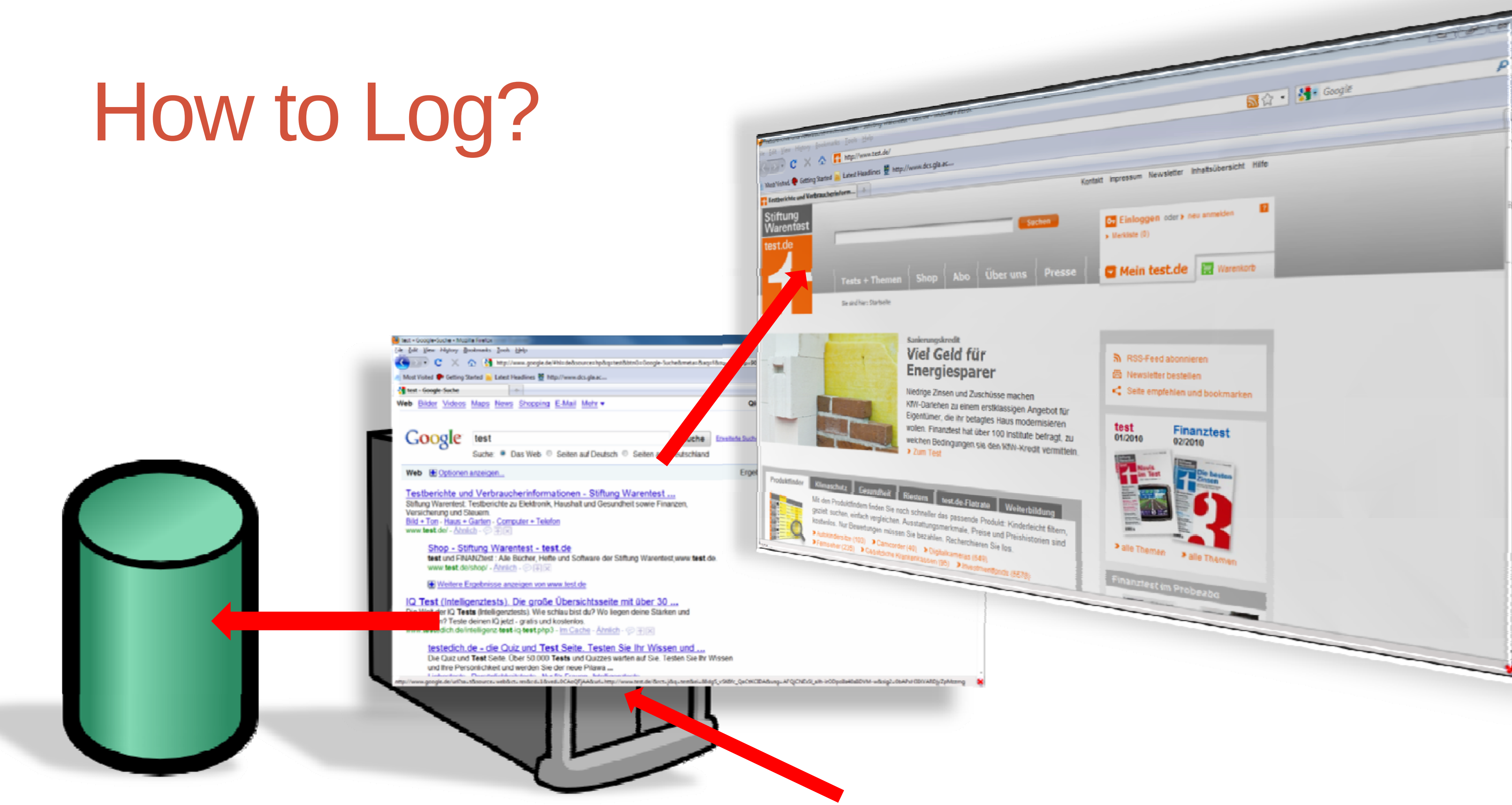
- Expensive
- Slowly
- Lack of Data

Logging

- Exists in huge amount
- Is free available
- Is always up-to-date

Google Logs Our Searches

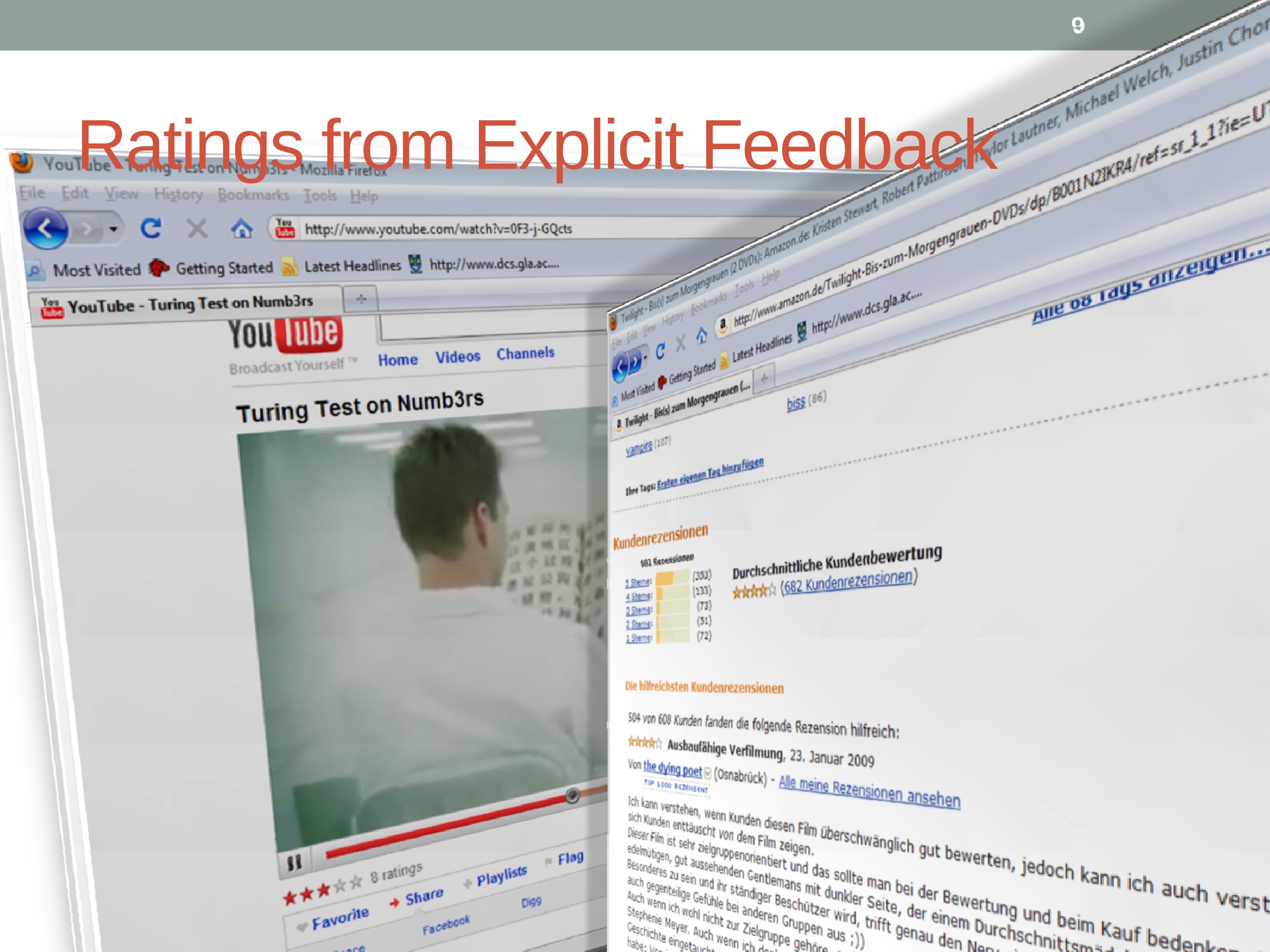
How to Log?



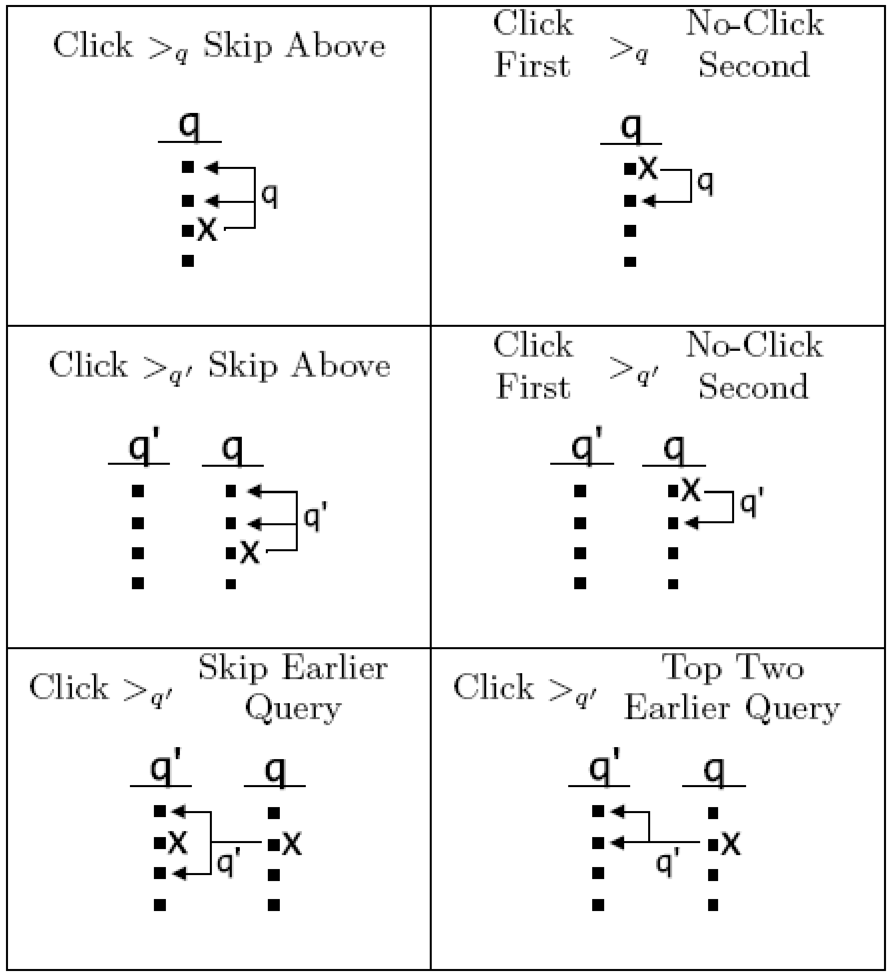
Feedback

- Ratings
- Preferences
- Rankings

Ratings from Explicit Feedback



Preferences from Clickthrough Rules



An Example

Google
lucene index browser
Search
Web
Show options...
Lucene Index Toolbox
these changes merited a slight change in name, from "Lucene Index Browser"
Toolbox", as this seems to better reflect the ...
Cached - Similar
Luke - a lucene index browser - Dev University - Wiki.Directi
13 Jun 2008 ... Luke - a lucene index browser. Luke is a handy development and diagnostic
tool, which accesses already existing Lucene indexes and allows ...
wiki.directi.com/pages/viewpage.action?pageId=3866631 - Cached - Similar
Luke - Lucene Index Browser
to the browser. At last save this page as Passing-Arrays-
Cached - Similar
Open Source, mature and high-performance Java search engine. ...
www.getopt.org/ - Cached - Similar
Apache Lucene
Features
Releases
Developers
Solr
Welcome to Lucene!
Search engine library with
phrase, and span queries,
Java - Solr - Mirrors - Apach
lucene.apache.org/
Doug Cutting. It is supported
en.wikipedia.org/wiki/Lucene

Rankings

- The more often a result is clicked, the better it is

Learning to Rank

- TF-IDF
- PageRank
- Machine learning approaches

Machine Learning Approaches for Learning to Rank

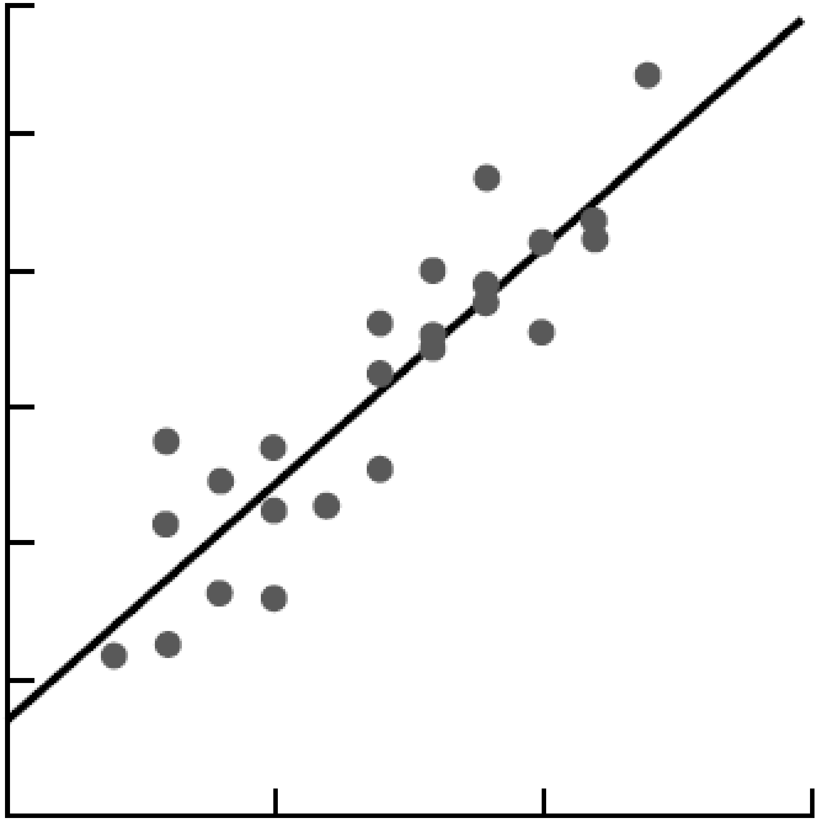
- Point-wise
  - Linear Regression
- Pair-wise
  - RankingSVM
  - MPRank
- List-wise
  - RBA (learns from single clicks)
  - Cofi-Rank

Feedback

- Rating
- Preferences
- Ranking

Linear Regression

- Documents are represented as vectors
- Each point corresponds to a document
- A score function will be learnt



Linear Regression (cont.)

- Document as feature vector
  - $X_i = (1 \quad x_{i1} \quad x_{i2} \quad \cdots \quad x_{ip})$
- Score function
  - $y_i = X_i W$

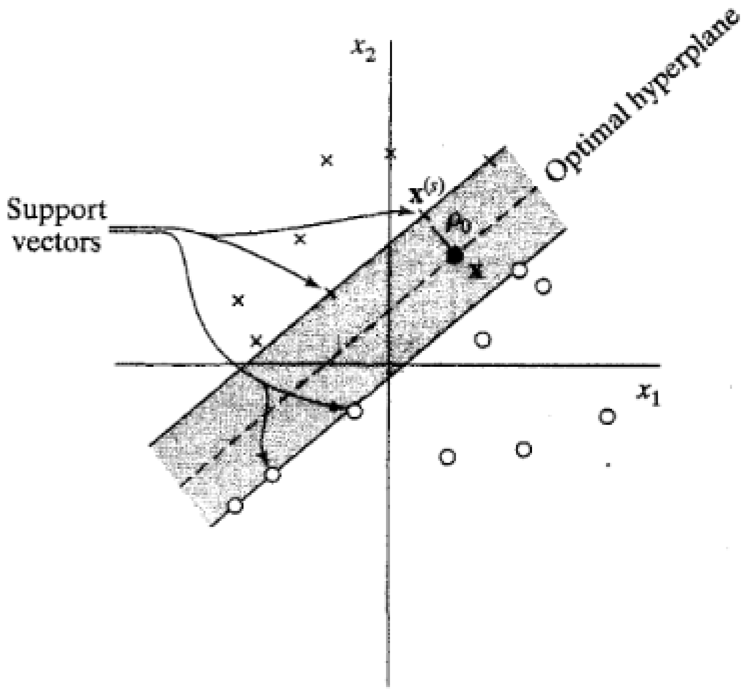
$$y_i = (1 \quad x_{i1} \quad \cdots \quad x_{ip}) \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{pmatrix}$$

- For all Query-Document-Pair
  - $Y = XW$
- $\widehat{W} = (X^T X)^{-1} X^T Y$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{pmatrix}$$

SVM

- Two classes $y \in \{+1, -1\}$ are separated by a hyperplane
  - $y_i(w^T x_i + b) \geq 1$
- Margin
  - $\rho = \frac{2}{\|w\|}$
- Misclassification are allowed
  - $y_i(w^T x_i + b) \geq 1 - \xi_i$
- To minimize:
  - $\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$



RankingSVM

- An application of SVM
- Constraint
  - $d_i >_q d_j \Leftrightarrow w\Phi(q, d_i) > w\Phi(q, d_j)$
- With misclassification
  - $w\left(\Phi(q, d_i) - \Phi(q, d_j)\right) > 1 - \xi_{ij}$
- Cost function
  - $\frac{1}{2} w^T w + C \sum_{i,j} \xi_{ij}$

MPRank

- Given score
  - y
- Score function
  - $h(x) = w\Phi(x)$
- Cost function for a pair of documents
  - $c^n{}_{MP} = |\big(h(x') - h(x)\big) - (y_{x'} - y_x)|^{\,n}$
- To minimize
  - $||w||^2 + C\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|\left(w\Phi(x_j) - w\Phi(x_i)\right) - (y_j - y_i)|^2$

RBA

Algorithm 2 Ranked Bandits Algorithm

initialize $\mathsf{MAB}_1(n), \ldots, \mathsf{MAB}_k(n)$ *Initialize MABs*
for t = 1 . . . T **do**
    for i = 1 . . . k **do** *Sequentially select documents*
        $\hat{b}_i(t) \leftarrow$ select-arm (MAB_i)
        if $\hat{b}_i(t) \in \{b_1(t), .., b_{i-1}(t)\}$ **then** *Replace repeats*
            $b_i(t) \leftarrow$ arbitrary unselected document
        else
            $b_i(t) \leftarrow \hat{b}_i(t)$
        end if
    end for
    display $\{b_1(t), \ldots, b_k(t)\}$ to user; record clicks
    for i = 1 . . . k **do** *Determine feedback for* MAB_i
        if user clicked $b_i(t)$ **and** $\hat{b}_i(t) = b_i(t)$ **then**
            $f_{it} = 1$
        else
            $f_{it} = 0$
        end if
        update $(\mathsf{MAB}_i, arm = \hat{b}_i(t), reward = f_{it})$
    end for
end for

Cofi-Rank

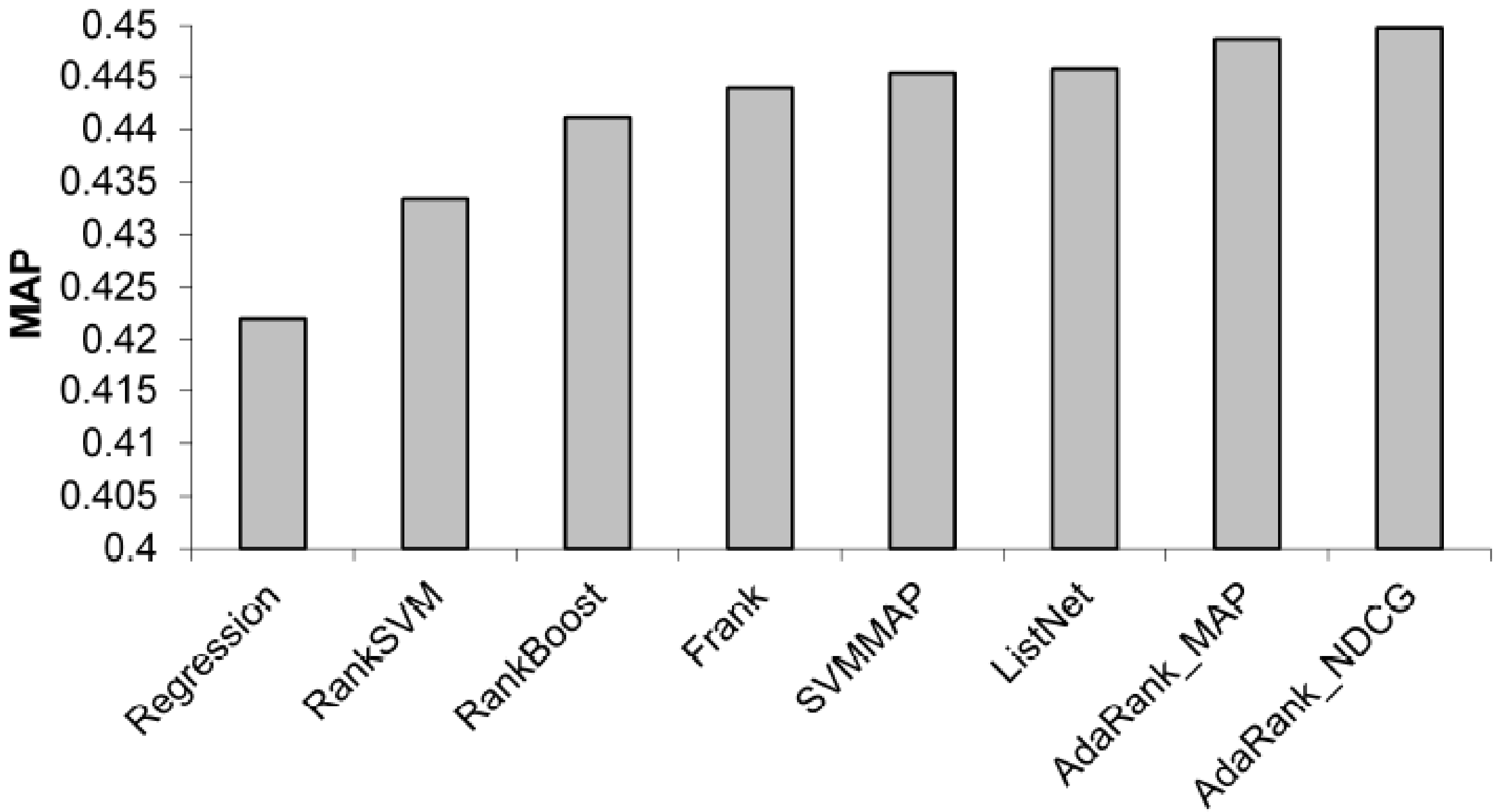
- 1st cost function
  - $\Delta(\pi, y) := 1 - NDCG(\pi, y)$
- 2nd cost function
  - $\psi(\pi, f) :=< c, f_{\pi} >$
- Upper bound
  - $l(f, y) := \max_{\pi}[\Delta(\pi, y) +< c, f_{\pi} - f >]\ s.t.$
  - $l(f, y) \geq \Delta(\pi^{*}, y) +< c, f_{\pi^{*}} - f >\geq \Delta(\pi^{*}, y)$
- Lost function over all the queries
  - $L(F, Y) := \sum_{i=1}^{u} l(F^{i}, Y^{i})$

- Ranking
  - π
- Score
  - y
- Score function
  - f

LETOR

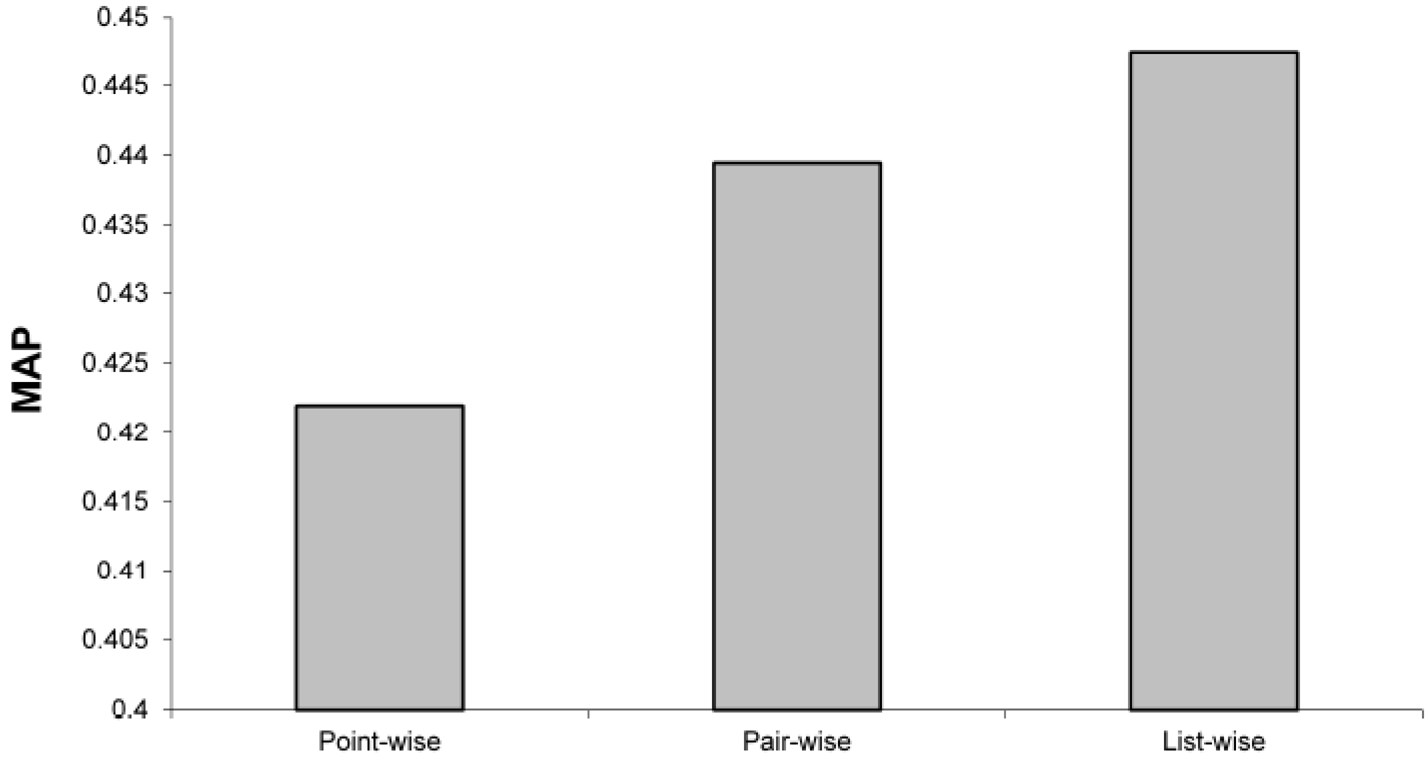
- A Benchmark Dataset introduced by Microsoft
  - Document => Vectors of features
  - Score is also given

Performance of Selected Approaches



- Data source: LETOR3.0 datasets

Mean Performance of Learning to Rank Type



- Data source: LETOR3.0 datasets

Advantages & Disadvantages

- Point-wise
  - Explicit feedback from users' score
- Pair-wise
  - Implicit feedback
  - In large quantities
  - Average quality
- List-wise
  - Best quality
  - Difficult to apply

Conclusion

- List-wise approaches achieve a better performance than point-wise and pair-wise approaches
- How to win training data from implicit feedback is a crucial problem

Thanks!

Q&A

References

- C. Cortes, M. Mohri, and A. Rastogi. Magnitude-preserving ranking algorithms. In ICML ’07: Proceedings of the 24th international conference on Machine learning, pages 169–176, New York, NY, USA, 2007. ACM.
- S. Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- M. H. Kutner, C. J. Nachtsheim, and J. Neter. Applied Linear Regression Models. McGraw-Hill/Irwin, fourth international edition, September 2004.

References(cont.)

- T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In LR4IR 2007, in conjunction with SIGIR 2007, 2007.
- F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 239-248, New York, NY, USA, 2005. ACM.
- F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multiarmed bandits. In ICML ’08: Proceedings of the 25th international conference on Machine learning, pages 784–791, New York, NY, USA, 2008. ACM.

References(cont.)

- F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In ICML ’08: Proceedings of the 25th international conference on Machine learning, pages 1192–1199, New York, NY, USA, 2008. ACM.