

Learning to Rank

CS6780 – Advanced Machine Learning
Spring 2019

Thorsten Joachims
Cornell University

Learning To Rank

Goal: Learn policy $\pi(x)$ that produces a ranking y of candidates w.r.t. query x .



Evaluating Rankings

- What influences the quality of a ranking?
 - How relevant is document d at rank r ?
 - Explicit feedback
 - User ratings
 - Expert ratings
 - Implicit feedback
 - Clicks, dwell time, mousing, scrolling, ...
 - How likely is user going to view rank r ?
 - Behavioral user model

Eye-Tracking

Eye tracking device



Record where and what people look at

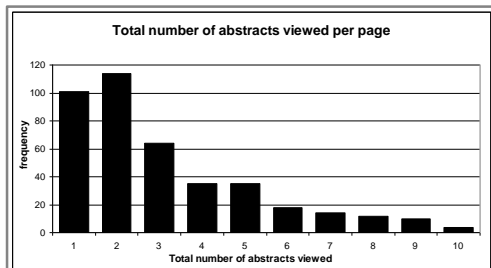
- **Fixations:** ~200-300ms; information is acquired
- **Saccades:** extremely rapid movements between fixations
- **Pupil dilation:** size of pupil indicates interest, arousal



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

[Granka et al., 2007]

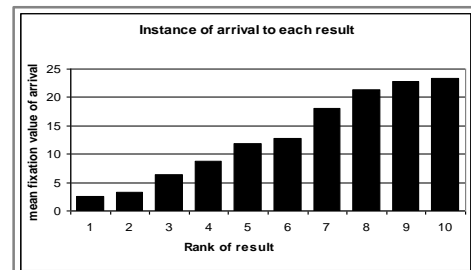
How Many Links do Users View?



Mean: 3.07 Median/Mode: 2.00

[Granka et al., 2007]

In Which Order are the Results Viewed?



=> Users tend to read the results in order

[Granka et al., 2007]

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)

[Granka et al., 2007]

Ranking Evaluation Metrics

Given: vector of relevance labels r

- Precision@k
 - Percentage of relevant results in top k
- Rank of Relevant documents

$$\Delta(y|r) = \sum_i \text{rank}(i|y) \cdot r_i$$

- Discounted Cumulative Gain (DCG)

$$\Delta(y|r) = \sum_i \frac{r_i}{\log(1 + \text{rank}(i|y))}$$

Learning to Rank Methods

- Joint feature map $\phi(x, d)$
 - Feature vector describing the match between query x and document d
- Pointwise LTR
 - Learn regression $\hat{r}: \phi(x, d) \rightarrow \mathcal{R}$
 - Prediction via $y = \text{argsort}\{\hat{r}(\phi(x, d))\}$
- Listwise LTR
 - Learn ranking policy π
 - Risk $R(\pi) = \int \Delta(\pi(x)|r) P(x, r)$
 - Minimize Empirical Risk $\hat{R}(\pi) = \sum_{(x,r)} \Delta(\pi(x)|r)$

Ranking SVM

- Data: $S = (x_j, D_j, r_j)^n$
- Policies: $y = \text{argsort}\{w \cdot \phi(x, d)\}$
- Training QP:

$$w^* = \underset{w, \xi \geq 0}{\text{argmin}} \frac{1}{2} w \cdot w + \frac{C}{n} \sum_j \sum_{(d, \bar{d})} \xi_j^{(d, \bar{d})}$$

$$\forall (d_1, \bar{d}_1) \in D_1: w \cdot [\phi(x_1, d_1) - \phi(x_1, \bar{d}_1)] \geq 1 - \xi_1^{(d, \bar{d})}$$

$$\vdots$$

$$\forall (d_n, \bar{d}_n) \in D_n: w \cdot [\phi(x_1, d_n) - \phi(x_1, \bar{d}_n)] \geq 1 - \xi_1^{(d, \bar{d})}$$

- Loss Bound:

$$\forall w: \text{rank}(d, \text{argsort}(w \cdot \phi(x, d))) \leq \sum_i \xi_i + \#\text{rel}$$

[Herbrich et al., 1999] [Joachims et al., 2002] [Joachims et al., 2007]

Explicit vs. Implicit Feedback

Explicit feedback

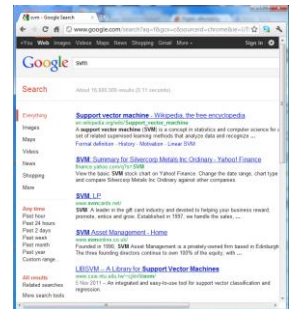
- Need to pay "experts"
- Slow to gather
- Potential expert-user mismatch
- Not personalized
- Complete feedback

Implicit feedback

- Free as by-product of system use
- Immediately available
- User provided, but spam-able
- Personalized
- Partial and biased by presentation

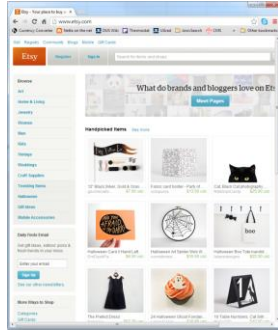
Interaction Logs: Search Engine

- Context x :
 - Query
- Action y :
 - Ranking
- Feedback $\delta(x, y)$:
 - Clicks on SERP



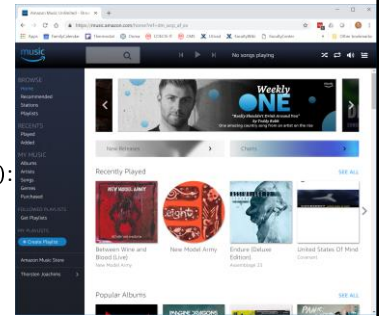
Interaction Logs: Online Retail

- Context x :
 - Category
- Action y :
 - Tile Layout
- Feedback $\delta(x, y)$:
 - Attributable purchases

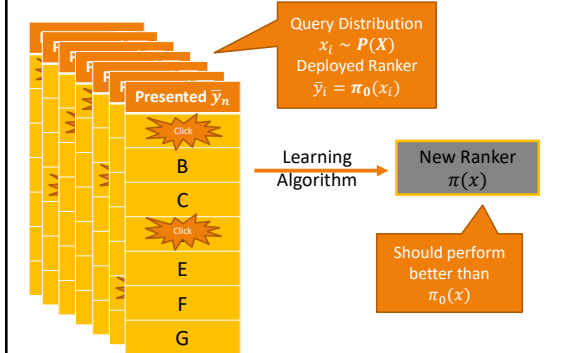


Interaction Logs: Streaming Media

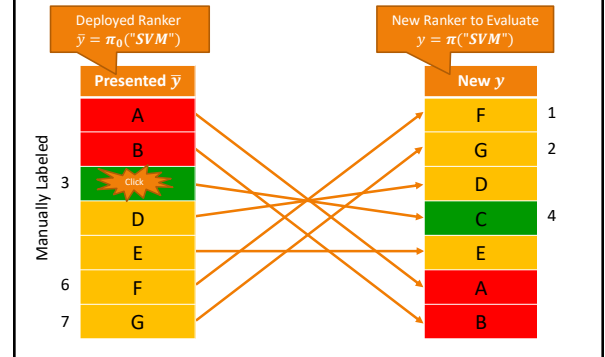
- Context x :
 - User
- Action y :
 - Tile layout
 - Scroll layout
- Feedback $\delta(x, y)$:
 - Plays



Learning-to-Rank from Clicks



Evaluating Rankings



Evaluation with Missing Judgments

- Loss: $\Delta(y|r)$
 - Relevance labels $r_i \in \{0,1\}$
 - This talk: rank of relevant documents
$$\Delta(y|r) = \sum_i \text{rank}(i|y) \cdot r_i$$
 - Assume:
 - Click implies observed and relevant: $(c_i = 1) \leftrightarrow (o_i = 1) \wedge (r_i = 1)$
 - Problem:
 - No click can mean not relevant OR not observed $(c_i = 0) \leftrightarrow (o_i = 0) \vee (r_i = 0)$
- Understand observation mechanism

Presented \bar{y}
A
B
C (Click)
D
E
F
G

[Joachims et al., 2017]

Inverse Propensity Score Estimator

- Observation Propensities $Q(o_i = 1|x, \bar{y}, r)$
 - Random variable $o_i \in \{0,1\}$ indicates whether relevance label r_i for is observed
- Inverse Propensity Score (IPS) Estimator:

$$\hat{\Delta}(y|r, o) = \sum_{i:c_i=1} \frac{\text{rank}(i|y)}{Q(o_i = 1|\bar{y}, r)}$$

- Unbiasedness: $E[\hat{\Delta}(y|r, o)] = \Delta(y|r)$

Need to know the propensities only for relevant/clicked docs.

Presented \bar{y}	Q
A	1.0
B	0.8
C	0.5
D	0.2
E	0.2
F	0.2
G	0.1

[Horvitz & Thompson, 1992] [Rubin, 1983] [Zadrozny et al., 2003] [Langford, Li, 2009] [Joachims et al., 2017]

ERM for Partial-Information LTR

- Unbiased Empirical Risk:

$$\hat{R}_{IPS}(\pi) = \frac{1}{N} \sum_{(x, \bar{y}, r) \in S} \sum_{i: c_i=1} \frac{\text{rank}(i|\pi(x))}{Q(o_i = 1|\bar{y}, r)}$$

Consistent Estimator of True Performance

- ERM Learning:

$$\hat{\pi} = \underset{\pi}{\text{argmin}} [\hat{R}_{IPS}(\pi)]$$

Consistent ERM Learning

- Questions:

- How do we optimize this empirical risk in a practical learning algorithm?
- How do we define and estimate the propensity model $Q(o_i = 1|\bar{y}, r)$?

[Joachims et al., 2017]

Propensity-Weighted SVM Rank

- Data:

$$S = (x_j, d_j, D_j, q_j)^n$$

Query Clicked Others Propensity

Optimizes convex upper bound on unbiased IPS risk estimate!

- Training QP:

$$w^* = \underset{w, \xi \geq 0}{\text{argmin}} \frac{1}{2} w \cdot w + \frac{C}{n} \sum_j \frac{1}{q_j} \sum_i \xi_j^i$$

$$\forall d^i \in D_1: w \cdot [\phi(x_1, d_1) - \phi(x_1, d^i)] \geq 1 - \xi_1^i$$

$$\vdots$$

$$\forall d^i \in D_n: w \cdot [\phi(x_n, d_n) - \phi(x_n, d^i)] \geq 1 - \xi_n^i$$

- Loss Bound:

$$\forall w: \text{rank}(d, \text{sort}(w \cdot \phi(x, d))) \leq \sum_i \xi^i + 1$$

[Herbrich et al., 1999] [Joachims et al., 2002] [Joachims et al., 2017]

Position-Based Propensity Model

- Model:

$$P(c_i = 1 | r_i, \text{rank}(i|\bar{y})) = q_{\text{rank}(i|\bar{y})} \cdot [r_i = 1]$$

- Assumptions

- Examination only depends on rank
- Click reveals relevance if rank is examined

Presented \bar{y}	Q
A	q_1
B	q_2
C	q_3
D	q_4
E	q_5
F	q_6
G	q_7

[Richardson et al., 2007] [Chuklin et al., 2015] [Wang et al., 2016]

Experiments

- Yahoo Web Search Dataset

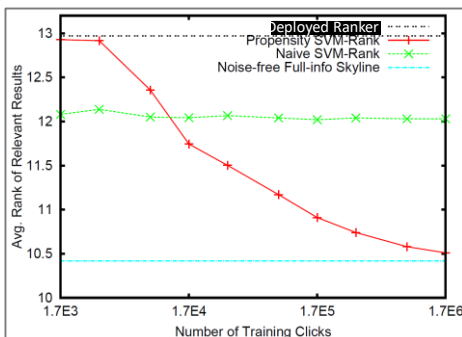
- Full-information dataset
- Binarized relevance labels

- Generate synthetic click data based on

- Position-based propensity model with $q_r = (\frac{1}{r})^\eta$
- Baseline "deployed" ranker to generate \bar{y}
- 33% noisy clicks on irrelevant docs

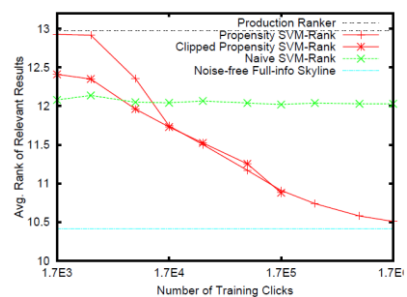
Presented \bar{y}	Q
A	q_1
B	q_2
Click	q_3
D	q_4
E	q_5
Click	q_6
G	q_7

Scaling with Training Set Size



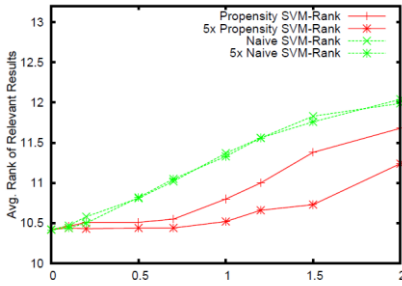
[Joachims et al., 2017]

Scaling with Training Set Size



[Joachims et al., 2017]

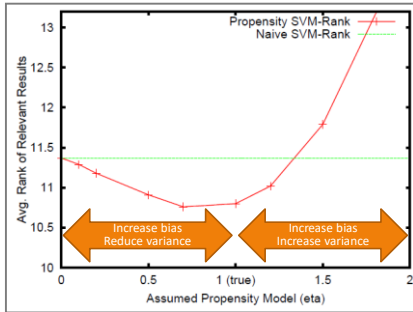
Severity of Presentation Bias



$$q_r = \left(\frac{1}{r}\right)^\eta$$

[Joachims et al., 2017]

Misspecified Propensities



$$q_r = \left(\frac{1}{r}\right)^\eta$$

[Joachims et al., 2017]

Position-Based Propensity Model

- Model:

$$P(c_i = 1 | r_i, \text{rank}(i|\bar{y})) = q_{\text{rank}(i|\bar{y})} \cdot [r_i = 1]$$

- Assumptions

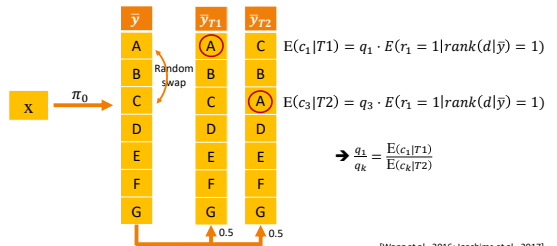
- Examination only depends on rank
- Click reveals relevance if rank is examined

Presented \bar{y}	q
A	q_1
B	q_2
C	q_3
D	q_4
E	q_5
F	q_6
G	q_7

[Richardson et al., 2007] [Chuklin et al., 2015] [Wang et al., 2016] [Joachims et al., 2017]

Estimating the Propensities

- Idea: Randomization to control for relevance
→ Swap Interventions



[Wang et al., 2016; Joachims et al., 2017]

Real-World Experiment

- Arxiv Full-Text Search
 - Run Swap(1,r) experiment to estimate q_r
 - Collect training clicks using production ranker
 - Train naïve / propensity SVM-Rank (1000 features)
 - A/B tests via interleaving

Interleaving Experiment	Propensity SVM-Rank		
	wins	loses	ties
against Prod	87	48	83
against Naive SVM-Rank	95	60	102



Conclusions

- Learning to Rank
 - from expert ratings
 - Pointwise: estimate relevance
 - Listwise: ERM to optimize ranking metric
 - from user interactions
 - Deal with missing relevance labels
 - Use IPS to get unbiased ERM objective
- Other Aspects
 - Fairness constraints on ranking policy