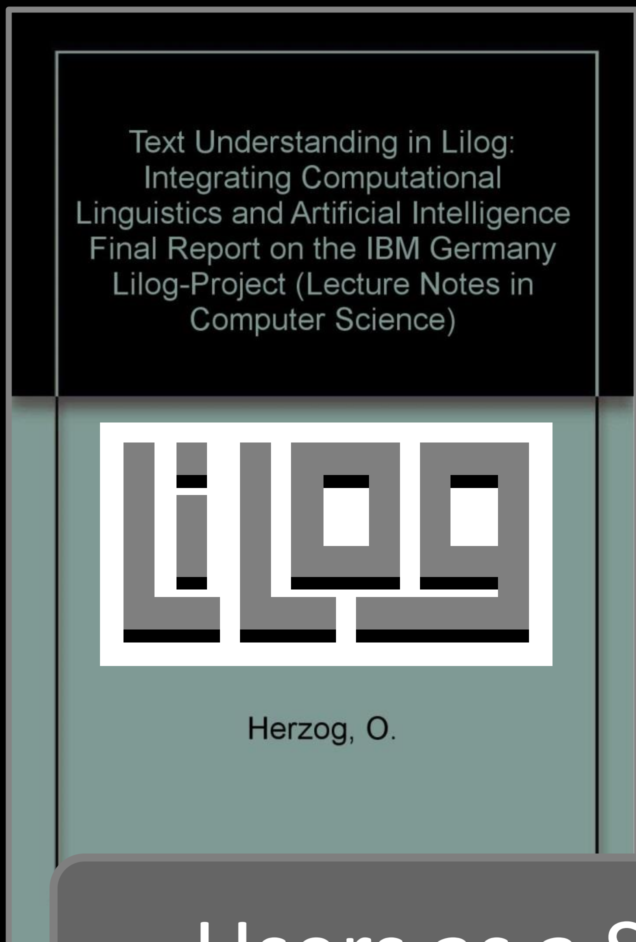
Learning from Rational* Behavior

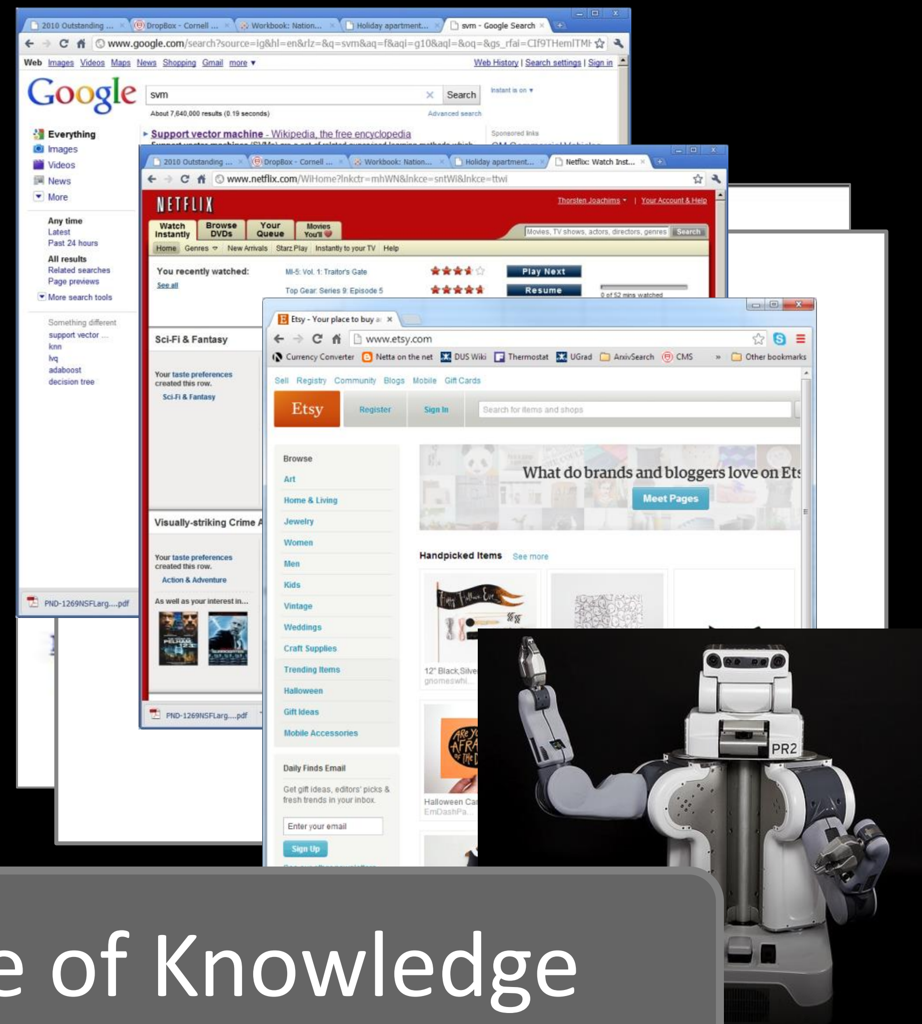
Josef Broder, Olivier Chapelle, Geri Gay, Arpita Ghosh, Laura Granka, Thorsten Joachims, Bobby Kleinberg, Madhu Kurup, Filip Radlinski, Karthik Raman, Tobias Schnabel, Pannaga Shivaswamy, Adith Swaminathan, Yisong Yue

Department of Computer Science
Department of Information Science
Cornell University

*) Restrictions apply. Some modeling required.

Knowledge-Based Systems





Users as a Source of Knowledge

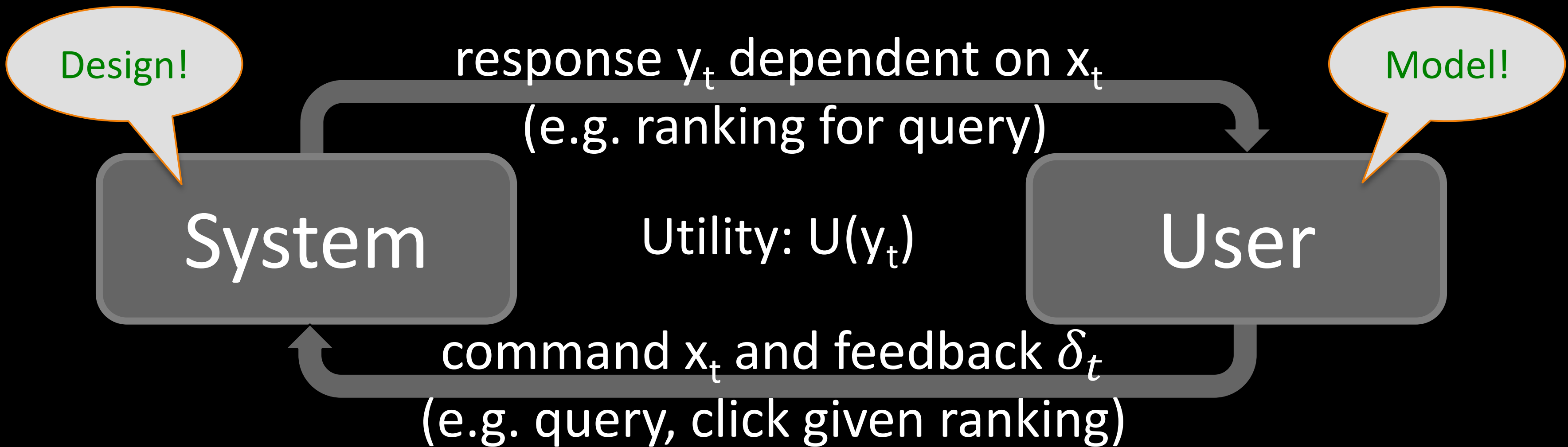
Supervised Batch Learning

- Data: $(x, y^*) \sim P(X, Y^*)$
  - x is input
  - y^* is true label
- Rules: $f \in H$
- Prediction: $\hat{y} = f(x)$
- Loss function: $\Delta(\hat{y}, y^*)$

→ Find $f \in H$ that minimizes prediction error

$$R(f) = \int \Delta(f(x), y^*)$$

Interactive Learning System



- Observed Data ≠ Training Data in Naïve Learning Model
  - Observed data is user's decisions
  - Decisions need proper interpretation
- New Learning Models that Integrate User and Algorithm
  - Decisions → Feedback → Learning Algorithm

Decide between two Ranking Functions

Distribution $P(x)$ of x=(user, query)

⋮
(tj,"SVM")
⋮

Retrieval Function 1
$f_1(x) \rightarrow y_1$

Which one is better?

Retrieval Function 2
$f_2(x) \rightarrow y_2$

Retrieval Function 1 results:

1. Kernel Machines
   http://svm.first.gmd.de/
2. SVM-Light Support Vector Machine
   http://svmlight.joachims.org/
3. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Service Master Company
   http://www.servicemaster.com/

⋮

U(tj,"SVM",y_1)

Retrieval Function 2 results:

1. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
2. Service Master Company
   http://www.servicemaster.com/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
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/

⋮

U(tj,"SVM",y_2)

Measuring Utility

Name	Description	Aggre-gation	Hypothesized Change with Decreased Quality
Abandonment Rate	% of queries with no click	N/A	Increase
Reformulation Rate	% of queries that are followed by reformulation	N/A	Increase
Queries per Session	Session = no interruption of more than 30 minutes	Mean	Increase
Clicks per Query	Number of clicks	Mean	Decrease
Click@1	% of queries with clicks at position 1	N/A	Decrease
Max Reciprocal Rank*	1/rank for highest click	Mean	Decrease
Mean Reciprocal Rank*	Mean of 1/rank for all clicks	Mean	Decrease
Time to First Click*	Seconds before first click	Median	Increase
Time to Last Click*	Seconds before final click	Median	Decrease

(*) only queries with at least one click count

ArXiv.org: User Study

User Study in ArXiv.org

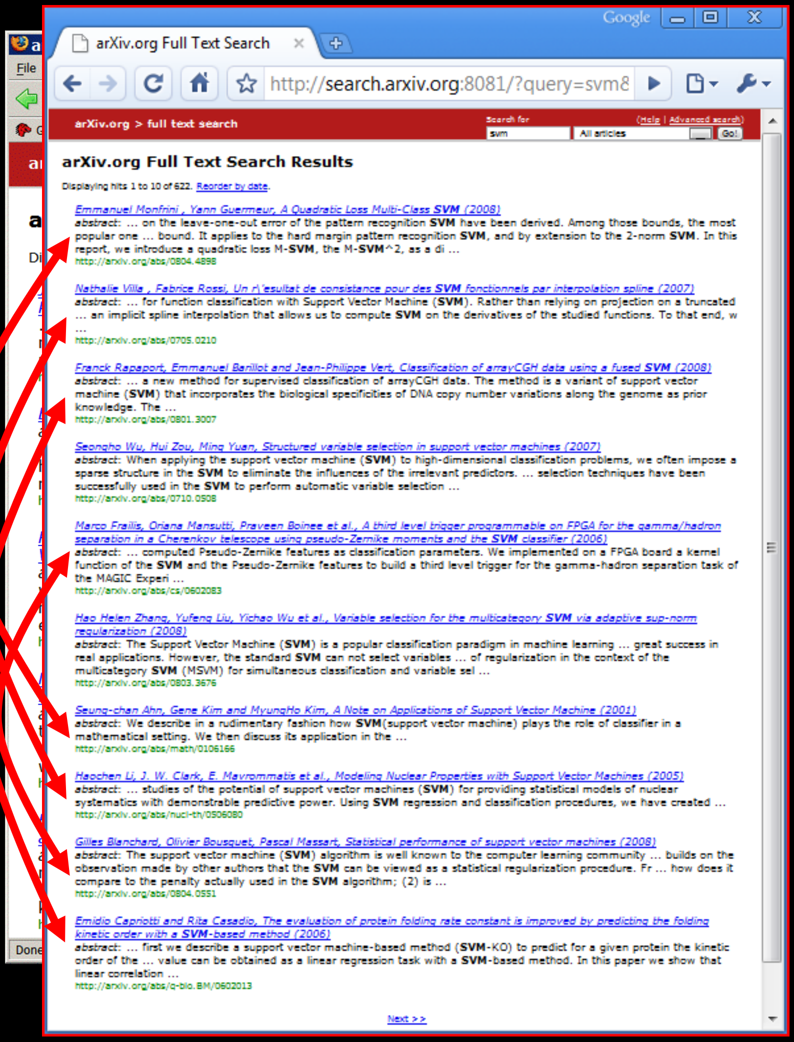
- Natural user and query population
- User in natural context, not lab
- Live and operational search engine
- Ground truth by construction

ORIG $\succ$ SWAP2 $\succ$ SWAP4

- ORIG: Hand-tuned fielded
- SWAP2: ORIG with 2 pairs swapped
- SWAP4: ORIG with 4 pairs swapped

ORIG $\succ$ FLAT $\succ$ RAND

- ORIG: Hand-tuned fielded
- FLAT: No field weights
- RAND : Top 10 of FLAT shuffled

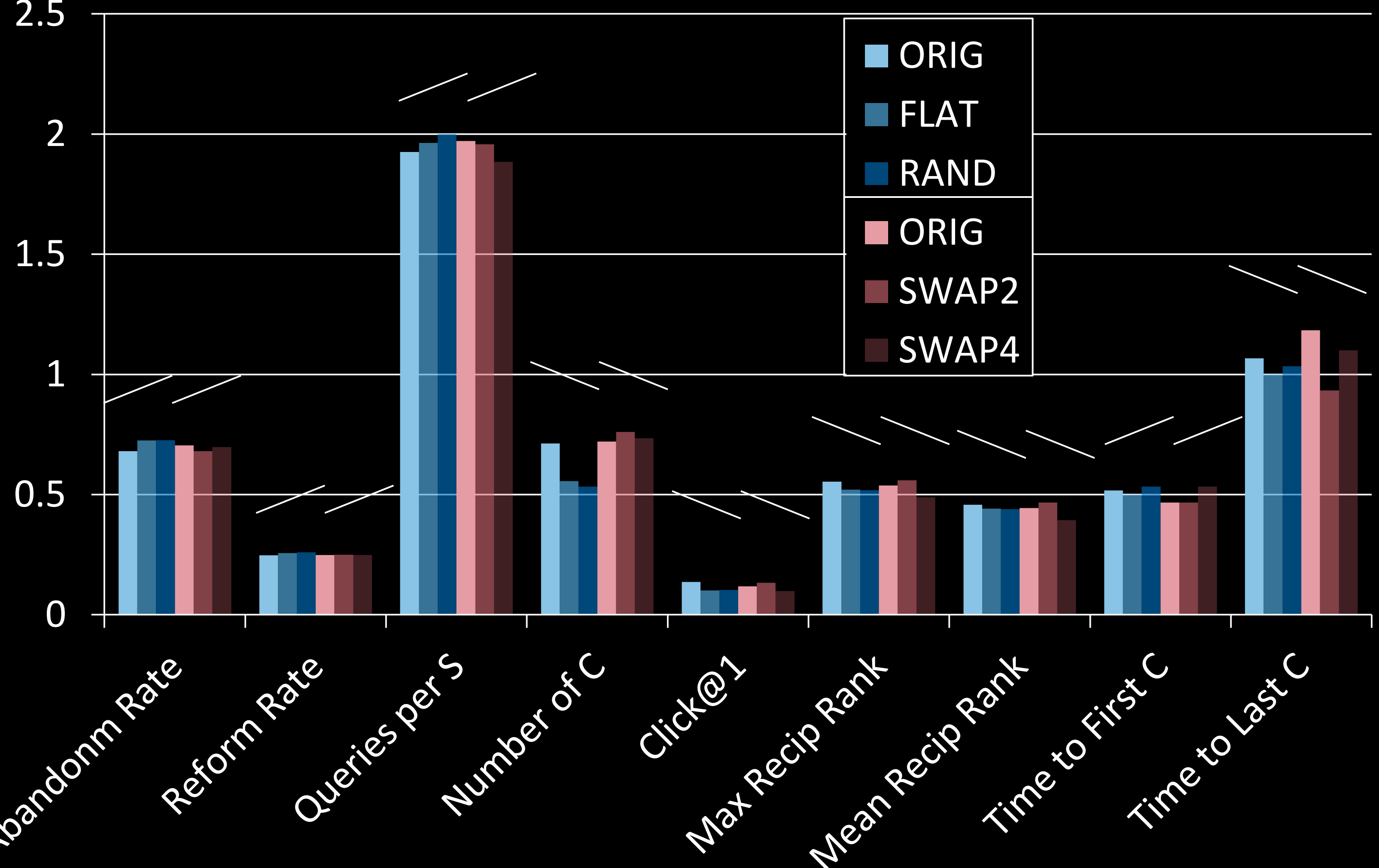


[Radlinski et al., 2008]

ArXiv.org: Experiment Setup

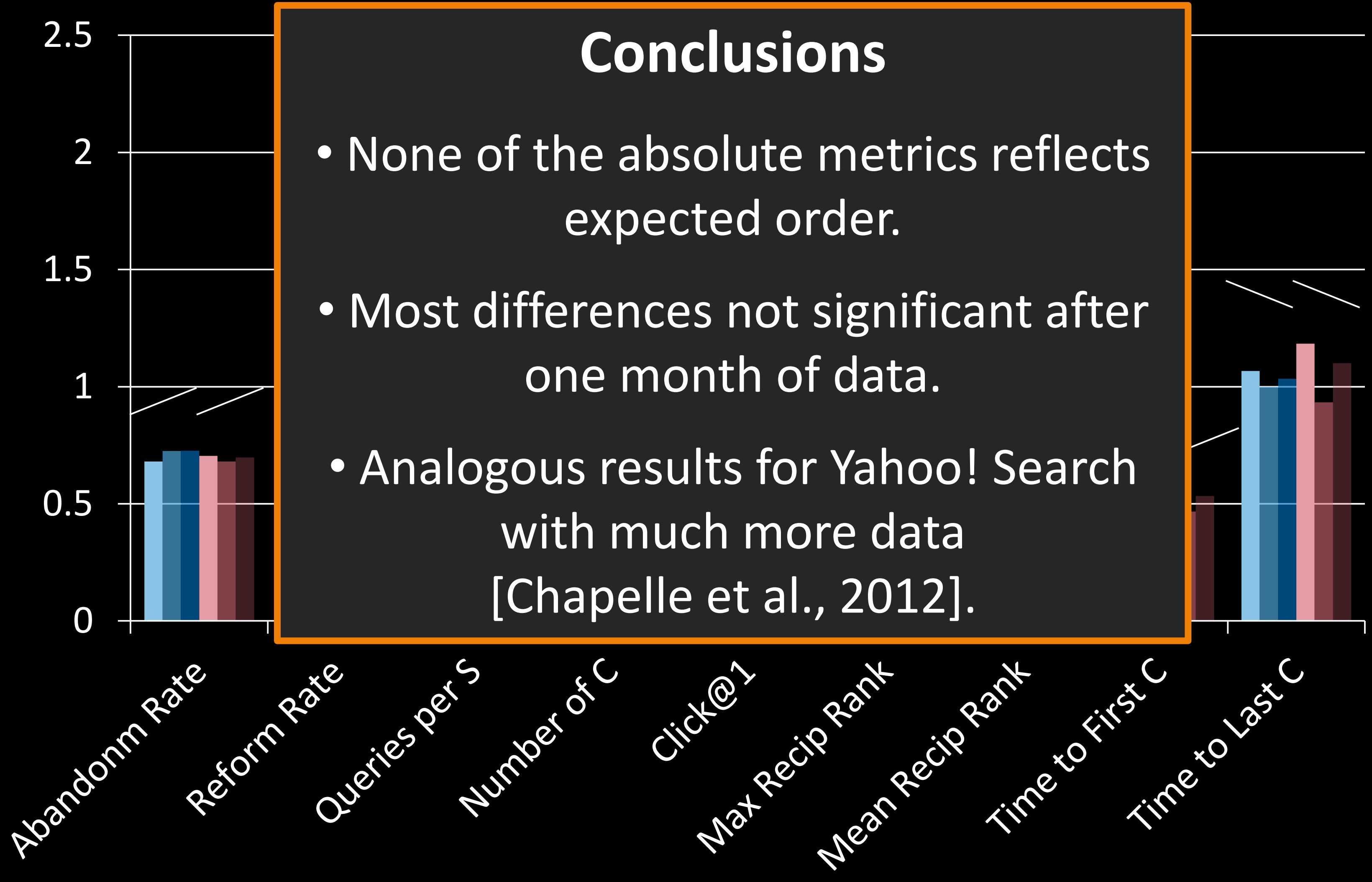
- Experiment Setup
  - Phase I: 36 days
    - Users randomly receive ranking from Orig, Flat, Rand
  - Phase II: 30 days
    - Users randomly receive ranking from Orig, Swap2, Swap4
  - User are permanently assigned to one experimental condition based on IP address and browser.
- Basic Statistics
  - ~700 queries per day / ~300 distinct users per day
- Quality Control and Data Cleaning
  - Test run for 32 days
  - Heuristics to identify bots and spammers
  - All evaluation code was written twice and cross-validated

Arxiv.org: Results



[Radlinski et al., 2008]

Arxiv.org: Results



Conclusions

- None of the absolute metrics reflects expected order.
- Most differences not significant after one month of data.
- Analogous results for Yahoo! Search with much more data [Chapelle et al., 2012].

[Radlinski et al., 2008]

How about Explicit Feedback?

35 of 36 people found the following review helpful

★★★☆☆ **a great buy for the price,a little hard to navigate for books**, November 13, 2012

By **flipnmelo** - See all my reviews

This review is from: Kindle, 6" E Ink Display, Wi-Fi - for international shipment (Electronics)

I purchased this for my fiance, and it's great overall, considering we live in a place with no bookstore and having to pay for shipping costs every time we wanted to read something was getting too much. With the Kindle we can download a book in no time.

What I don't like is that when searching for books under certain categories, there is no way to navigate to the end of the list any quicker. If I purchased all the books under the list on the first 5 pages, lets say, of comedy, and I'm ready to buy another I still have to manually go through all the pages of the list again, there's no way to skip ahead. But if we know what book we want, then it's worth it.

Also, the cost of some of the downloads are just as expensive as buying the books themselves. I thought that maybe it would cost less since there would be no costs for the print, paper, ink etc.

Help other customers find the most helpful reviews

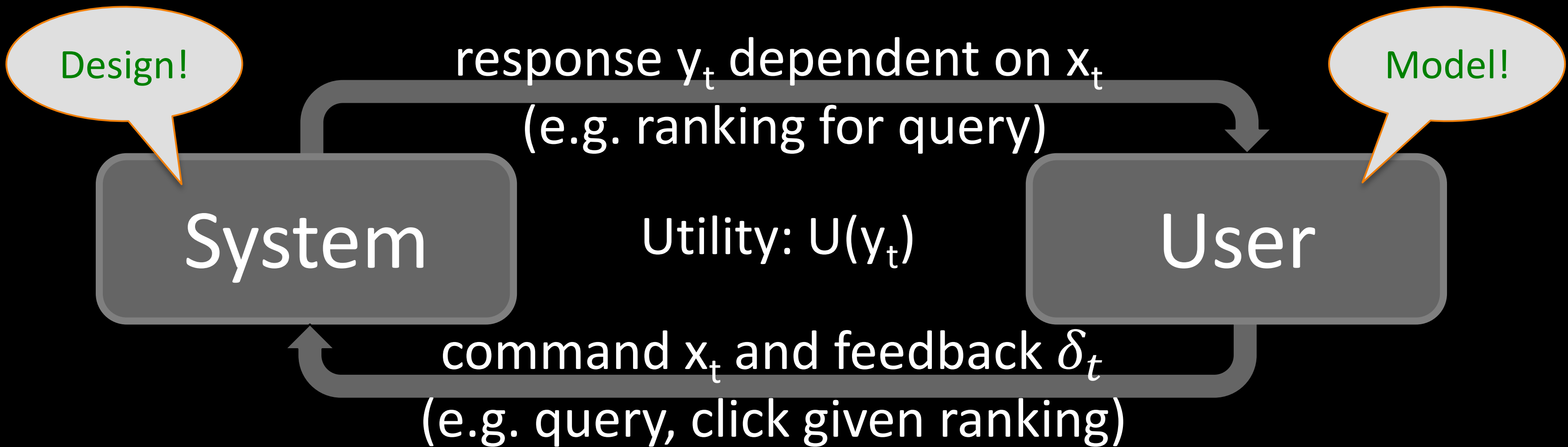
Was this review helpful to you? Yes No

Report abuse | Permalink

Comment

[Sipos et al., WWW14]

Interactive Learning System



- Observed Data ≠ Training Data ✓
- Decisions → Feedback → Learning Algorithm
  - Model the users decision process to extract feedback
  - Design learning algorithm for this type of feedback

Decide between two Ranking Functions

Distribution P(x) of x=(user, query)

⋮
(tj,"SVM")
⋮

Retrieval Function 1
$f_1(x) \rightarrow y_1$

Which one is better?

Retrieval Function 2
$f_2(x) \rightarrow y_2$

1. Kernel Machines
   http://svm.first.gmd.de/
2. SVM-Light Support Vector Machine
   http://svmlight.joachims.org/
3. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Service Master Company
   http://www.servicemaster.com/

⋮

$U(tj,\text{"SVM"},y_1)$

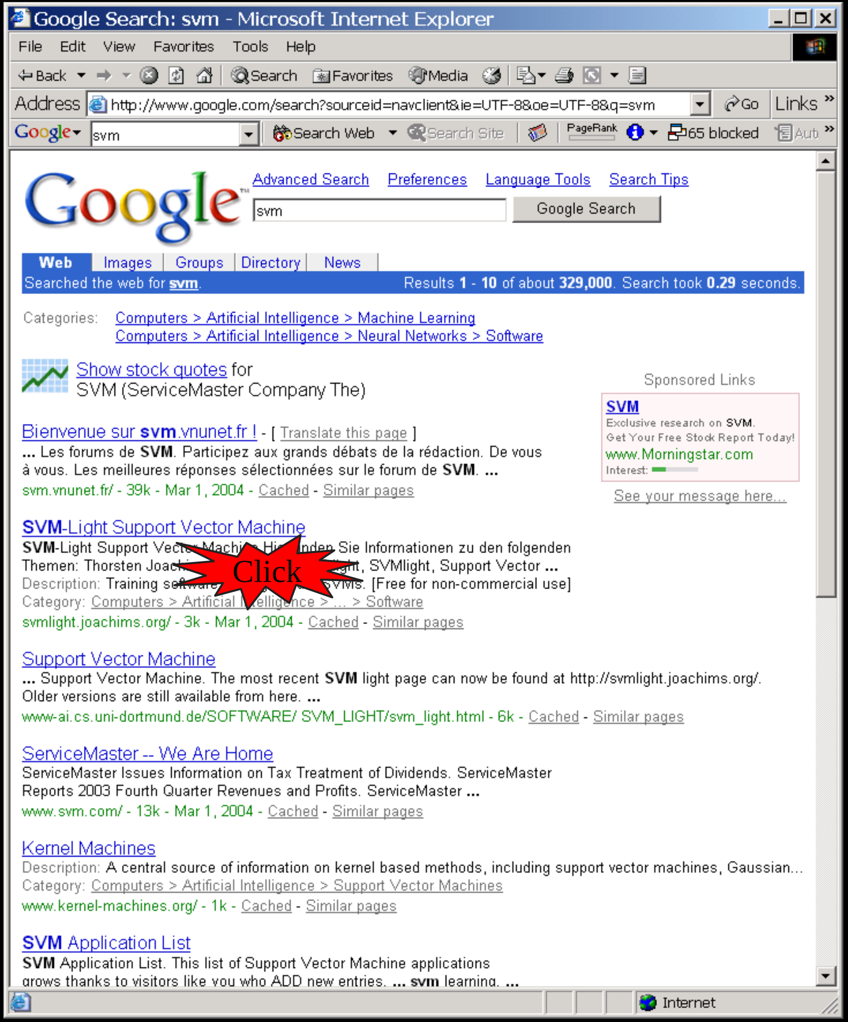
1. School of Veterinary Medicine at UPenn
   http://www.vet.upenn.edu/
2. Service Master Company
   http://www.servicemaster.com/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
4. Archives of SUPPORT-VECTOR-MACHINES
   http://www.jiscmail.ac.uk/lists/SUPPORT...
5. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm light/

⋮

$U(tj,\text{"SVM"},y_2)$

Economic Models of Decision Making

- Rational Choice
  - Alternatives $\mathcal{Y}$
  - Utility function U(y)
  - Decision $y^* = \mathrm{argmax}_{y \in \mathcal{Y}} \{U(y)\}$
- Bounded Rationality
  - Time constraints
  - Computation constraints
  - Approximate U(y)
- Behavioral Economics
  - Framing
  - Fairness
  - Loss aversion
  - Handling uncertainty



A Model of how Users Click in Search

- Model of clicking:
  - Users explore ranking to position k
  - Users click on most relevant (looking) links in top k
  - Users stop clicking when time budget up or other action more promising (e.g. reformulation)
  - Empirically supported by [Granka et al., 2004]

Balanced Interleaving

x=(u=tj, q="svm")

$f_1(x) \rightarrow y_1$

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/

$f_2(x) \rightarrow y_2$

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.*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

$\text{Interleaving}(y_1,y_2)$

1.	Kernel Machines *http://svm.first.gmd.de/*	**1**
2.	Support Vector Machine *http://jbolivar.freeservers.com/*	**2**
3.	SVM-Light Support Vector Machine *http://ais.gmd.de/~thorsten/svm light/*	**2**
4.	An Introduction to Support Vector Machines *http://www.support-vector.net/*	**3**
5.	Support Vector Machine and Kernel ... References *http://svm.*research.bell-labs.com/SVMrefs.html	**3**
6.	Archives of SUPPORT-VECTOR-MACHINES ... *http://www.jiscmail.ac.uk/lists/SUPPORT...*	**4**
7.	Lucent Technologies: SVM demo applet *http://svm.*research.bell-labs.com/SVT/SVMsvt.html	**4**

Model of User:
Better retrieval functions is more likely to get more clicks.

Invariant:
For all k, top k of balanced interleaving is union of top k_1 of r_1 and top k_2 of r_2 with $k_1=k_2 \pm 1$.

Interpretation: $(y_1 \succ y_2) \leftrightarrow \text{clicks}(\text{topk}(y_1)) > \text{clicks}(\text{topk}(y_2))$

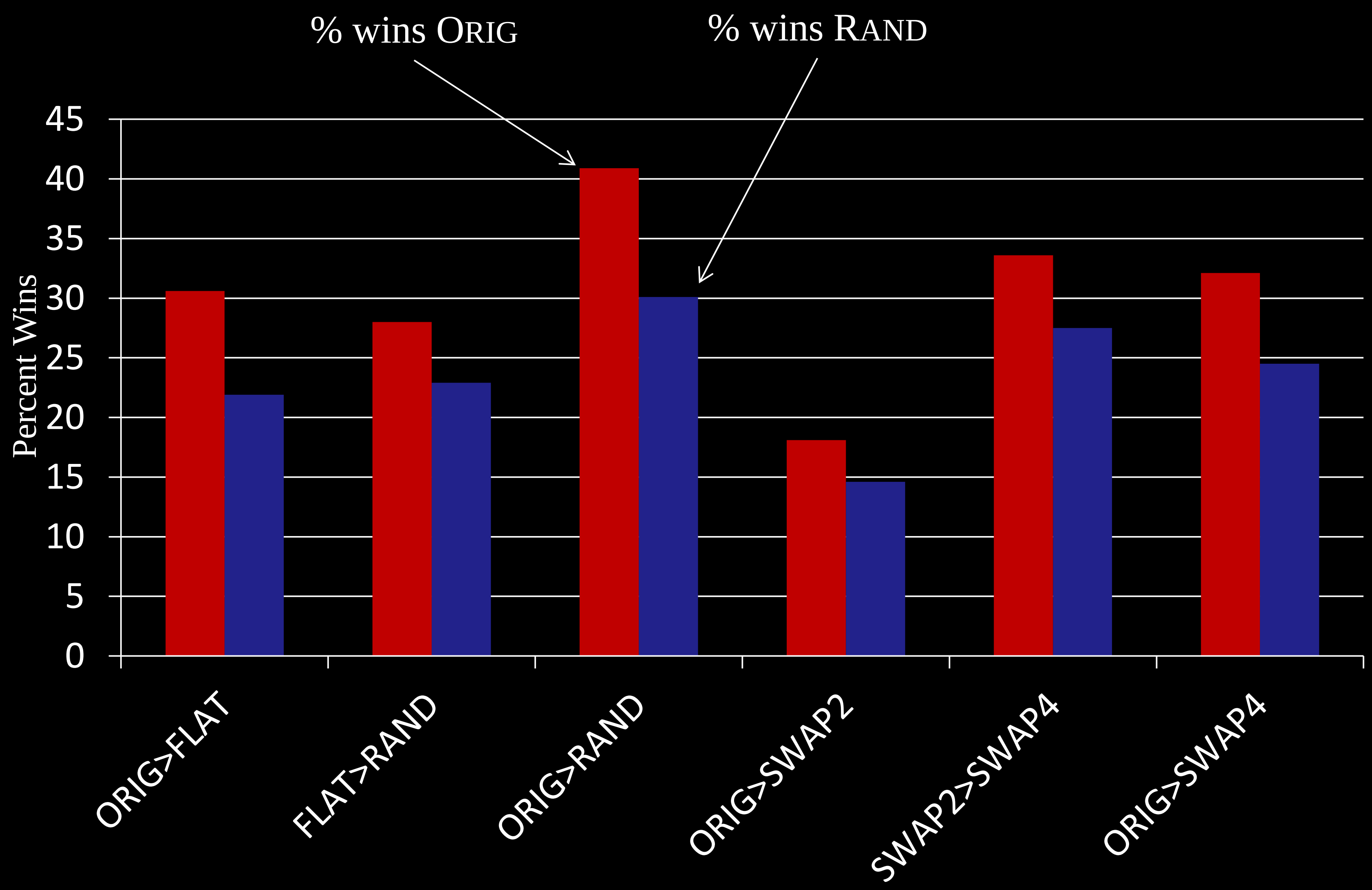
→ see also [Radlinski, Craswell, 2012] [Hofmann, 2012]

[Joachims, 2001] [Radlinski et al., 2008]

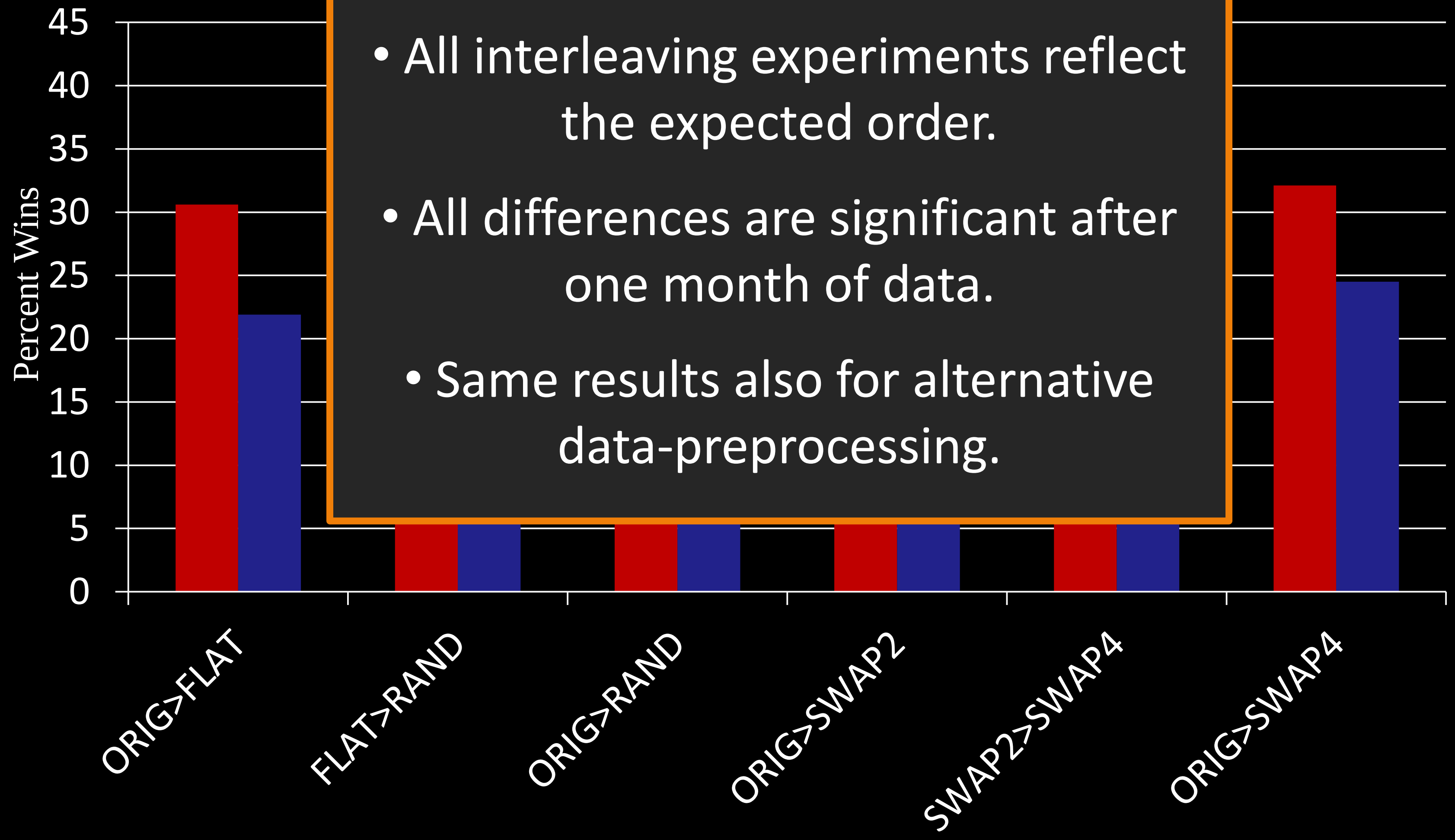
Arxiv.org: Interleaving Experiment

- Experiment Setup
  - Phase I: 36 days
    - Balanced Interleaving of (Orig,Flat) (Flat,Rand) (Orig,Rand)
  - Phase II: 30 days
    - Balanced Interleaving of (Orig,Swap2) (Swap2,Swap4) (Orig,Swap4)
- Quality Control and Data Cleaning
  - Same as for absolute metrics

Arxiv.org: Interleaving Results



Arxiv.org: Interleaving Results



Conclusions

- All interleaving experiments reflect the expected order.
- All differences are significant after one month of data.
- Same results also for alternative data-preprocessing.

Yahoo and Bing: Interleaving Results

- Yahoo Web Search [Chapelle et al., 2012]
  - Four retrieval functions (i.e. 6 paired comparisons)
  - Balanced Interleaving

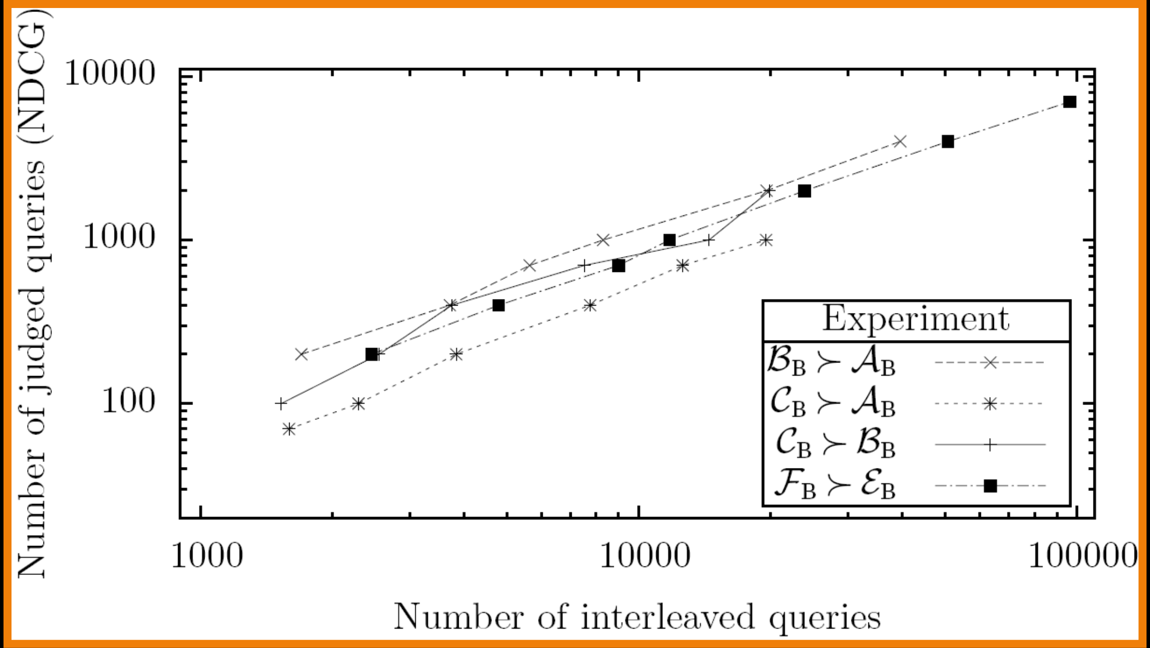
    → All paired comparisons consistent with ordering by NDCG.

- Bing Web Search [Radlinski & Craswell, 2010]
  - Five retrieval function pairs
  - Team-Game Interleaving

    → Consistent with ordering by NDGC when NDCG significant.

Efficiency: Interleaving vs. Explicit

- Bing Web Search
  - 4 retrieval function pairs
  - ~12k manually judged queries
  - ~200k interleaved queries

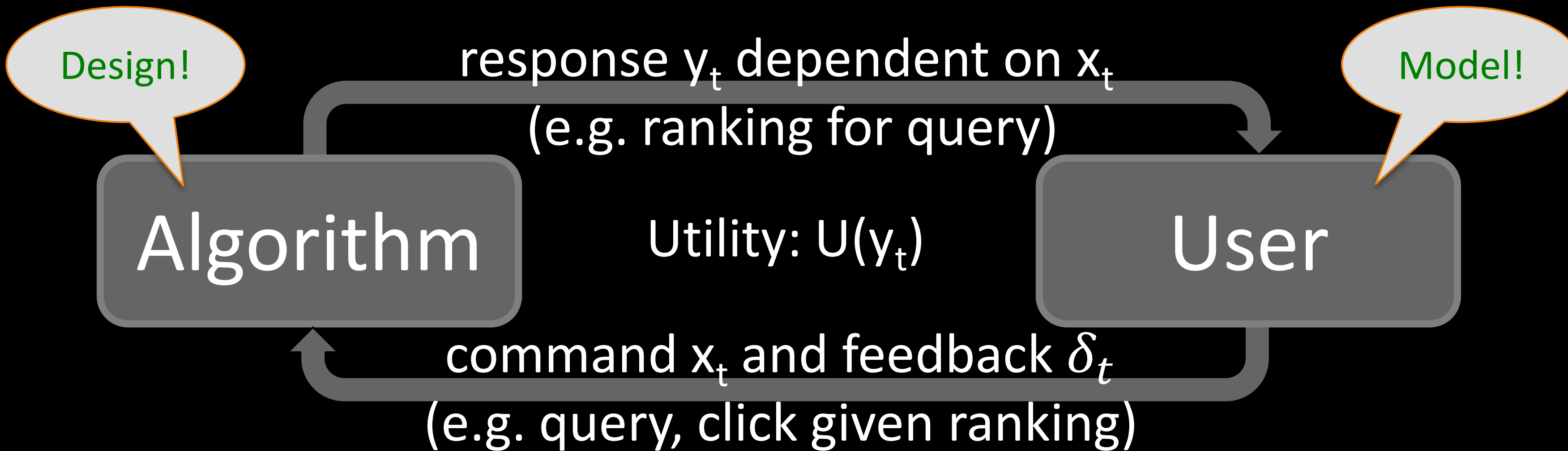


- Experiment
  - p = probability that NDCG is correct on subsample of size y
  - x = number of queries needed to reach same p-value with interleaving

➔ Ten interleaved queries are equivalent to one manually judged query.

[Radlinski & Craswell, 2010]

Interactive Learning System



- Observed Data ≠ Training Data ✓
- Decisions → Feedback → Learning Algorithm
  - Model the users decision process to extract feedback ✓
    → Pairwise comparison test $P(y_i \succ y_j \mid U(y_i)>U(y_j))$
  - Design learning algorithm for this type of feedback

Supervised Batch Learning

- Data: $(x, y^*) \sim P(X, Y^*)$
  - x is input
  - y^* is true label
- Rules: $f \in H$
- Prediction: $\hat{y} = f(x)$
- Loss function: $\Delta(\hat{y}, y^*)$

→ Find $f \in H$ that minimizes prediction error

$$R(f) = \int \Delta(f(x), y^*)$$

Learning on Operational System

- Example: 4 retrieval functions: A > B >> C > D
  - 10 possible pairs for interactive experiment
    - (A,B) → low cost to user
    - (A,C) → medium cost to user
    - (C,D) → high cost to user
    - (A,A) → zero cost to user
    - …
- Minimizing Regret
  - Don't present "bad" pairs more often than necessary
  - Trade off (long term) informativeness and (short term) cost
  - Definition: Probability of (f_t, f_t') losing against the best f^*

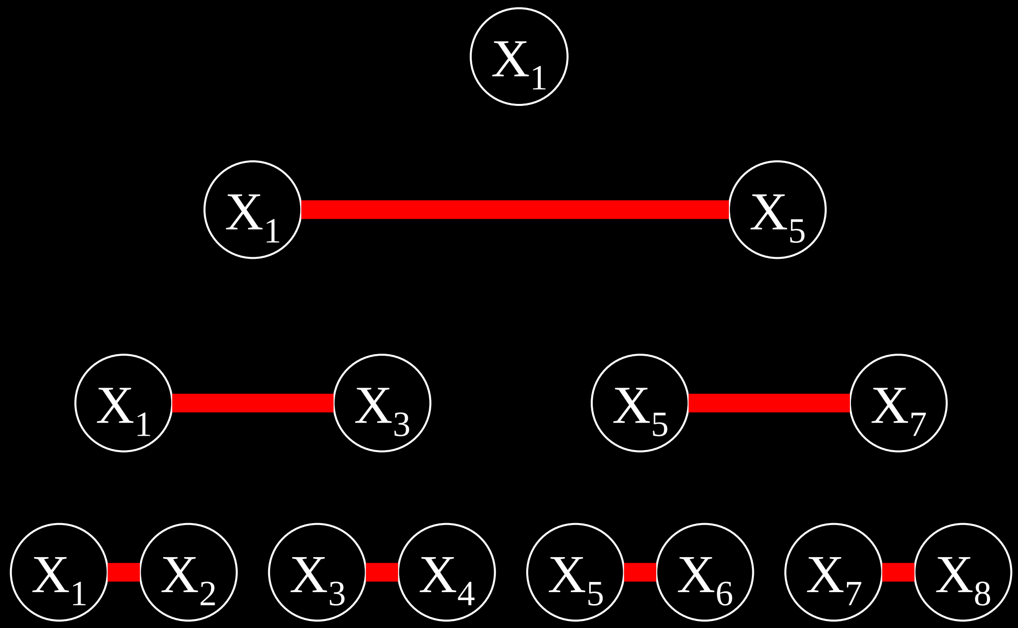
$$\mathrm{R(A)} = \sum_{t=1}^{T} [P(f^* \succ f_t) - 0.5] + [P(f^* \succ f_t') - 0.5]$$

→ Dueling Bandits Problem

[Yue, Broder, Kleinberg, Joachims, 2010]

First Thought: Tournament

- Noisy Sorting/Max Algorithms:
  - [Feige et al.]: Triangle Tournament Heap $O(n/\varepsilon^2 \log(1/\delta))$ with prob $1-\delta$
  - [Adler et al., Karp & Kleinberg]: optimal under weaker assumptions



Algorithm: Interleaved Filter 2

- Algorithm
  - InterleavedFilter1(T,W={f_1…f_K})
    - Pick random f’ from W
    - $\delta=1/(TK^2)$
    - WHILE |W|>1
      - FOR b ∈ W DO
        - » duel(f’,f)
        - » update P_f
      - t=t+1
      - $c_t=(\log(1/\delta)/t)^{0.5}$
      - Remove all f from W with $P_f < 0.5-c_t$ [WORSE WITH PROB $1-\delta$]
      - IF there exists f’’ with $P_{f''} > 0.5+c_t$ [BETTER WITH PROB $1-\delta$]
        - » Remove f’ from W
        - » Remove all f from W that are empirically inferior to f’
        - » f‘=f’’; t=0
    - UNTIL T: duel(f’,f’)

f_1	f_2	$f'=f_3$	f_4	f_5
0/0	0/0		0/0	0/0

f_1	f_2	$f'=f_3$	f_4	f_5
8/2	7/3		4/6	~~1/9~~

f_1	f_2	$f'=f_3$	f_4	
13/2	11/4	~~ ~~	~~7/8~~	XX

$f'=f_1$	f_2		f_4	
0/0	0/0	XX	XX	XX

Related Algorithms: [Hofmann, Whiteson, Rijke, 2011] [Yue, Joachims, 2009] [Yue, Joachims, 2011]

[Yue et al., 2009]

Assumptions

- Preference Relation: $f_i \succ f_j \Leftrightarrow P(f_i \succ f_j) = 0.5+\varepsilon_{i,j} > 0.5$
- Weak Stochastic Transitivity: $f_i \succ f_j$ and $f_j \succ f_k \rightarrow f_i \succ f_k$

Theorem: IF2 incurs expected average regret bounded by

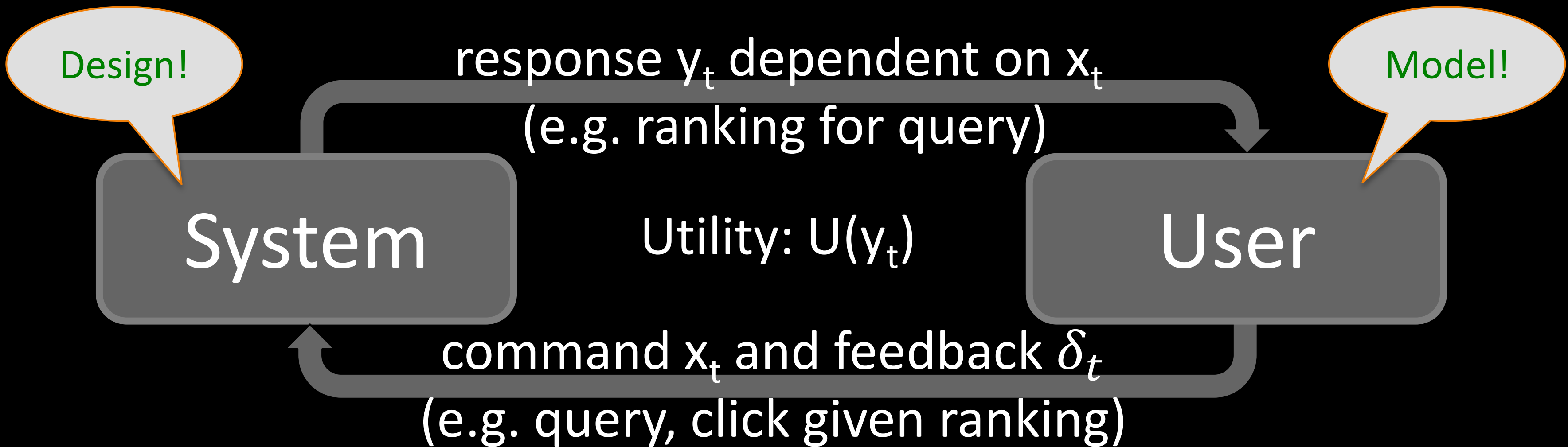
$$\frac{1}{T}E(R_T) \leq O\left(\frac{K}{\epsilon_{1,2}}\frac{\log T}{T}\right)$$

- S
- Stochastic Triangle Inequality: $f_i \succ f_j \succ f_k \rightarrow \varepsilon_{i,k} \leq \varepsilon_{i,j}+\varepsilon_{j,k}$

  $\varepsilon_{1,2} = 0.01$ and $\varepsilon_{2,3} = 0.01 \rightarrow \varepsilon_{1,3} \leq 0.02$
- ε-Winner exists: $\varepsilon = \max_i\{ P(f_1 \succ f_i)-0.5 \} = \varepsilon_{1,2} > 0$

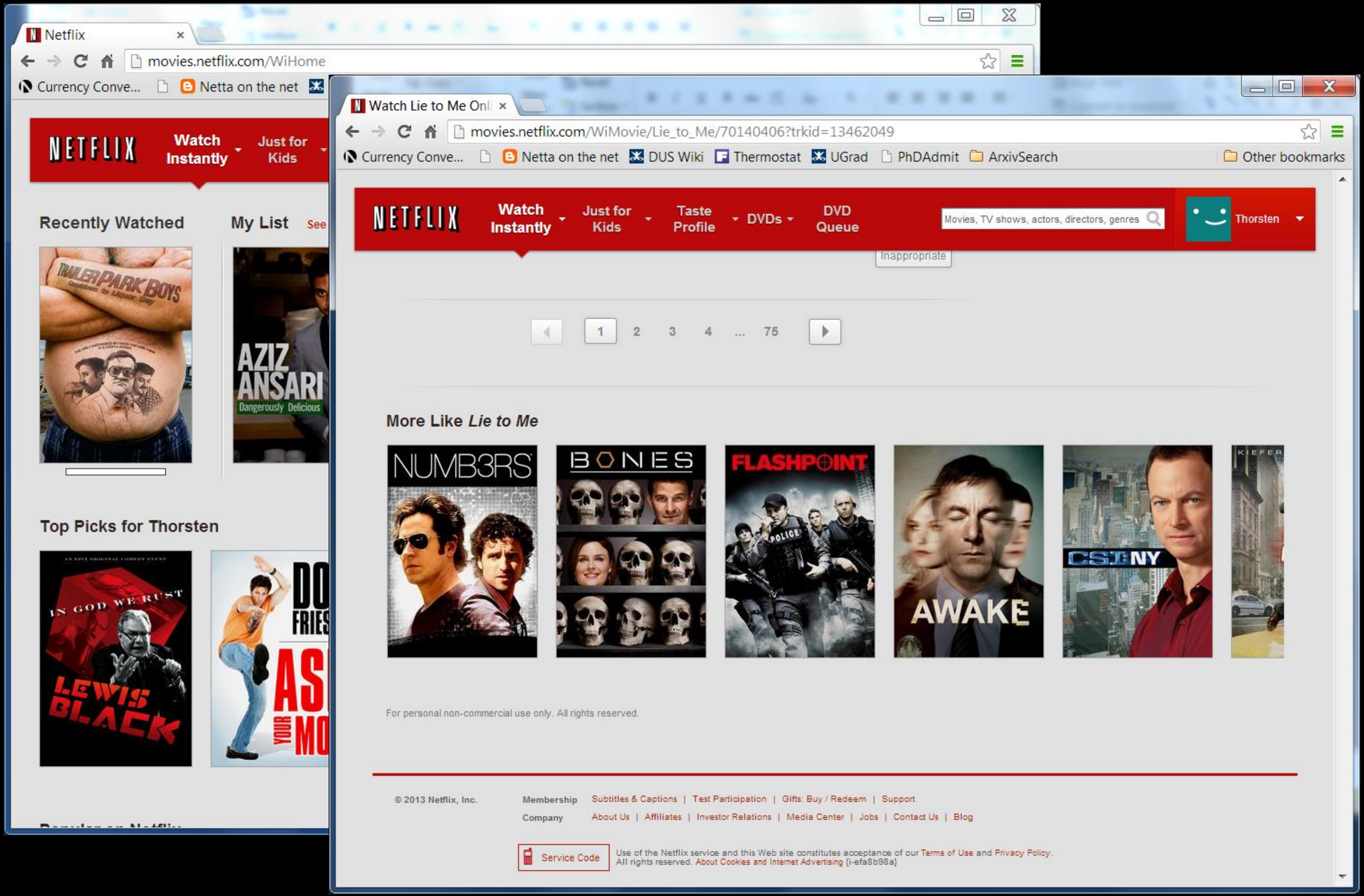
[Yue et al., 2009]

Interactive Learning System

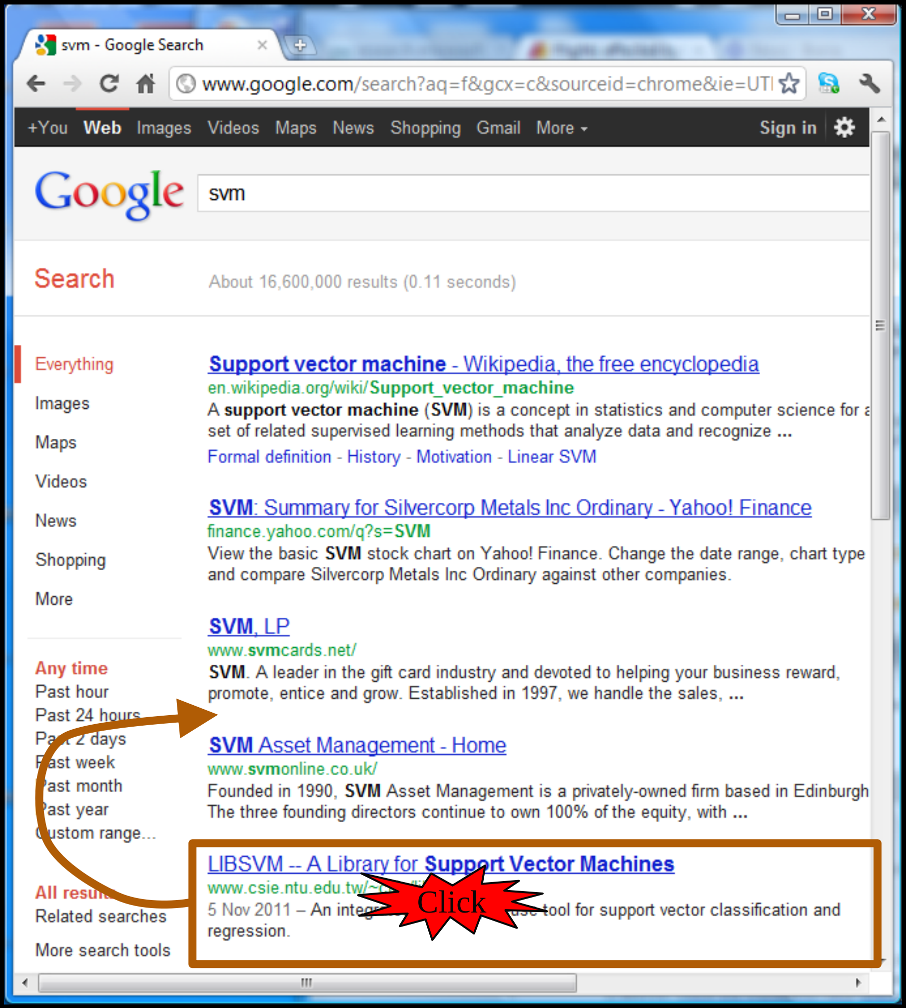


- Observed Data ≠ Training Data ✓
- Decisions → Feedback → Learning Algorithm
  - Model the users decision process to extract feedback ✓
    → Pairwise comparison test $P(\, y_i \succ y_j \mid U(y_i)>U(y_j)\,)$
  - Design learning algorithm for this type of feedback ✓
    → Dueling Bandits problem and algorithms (e.g. IF2)

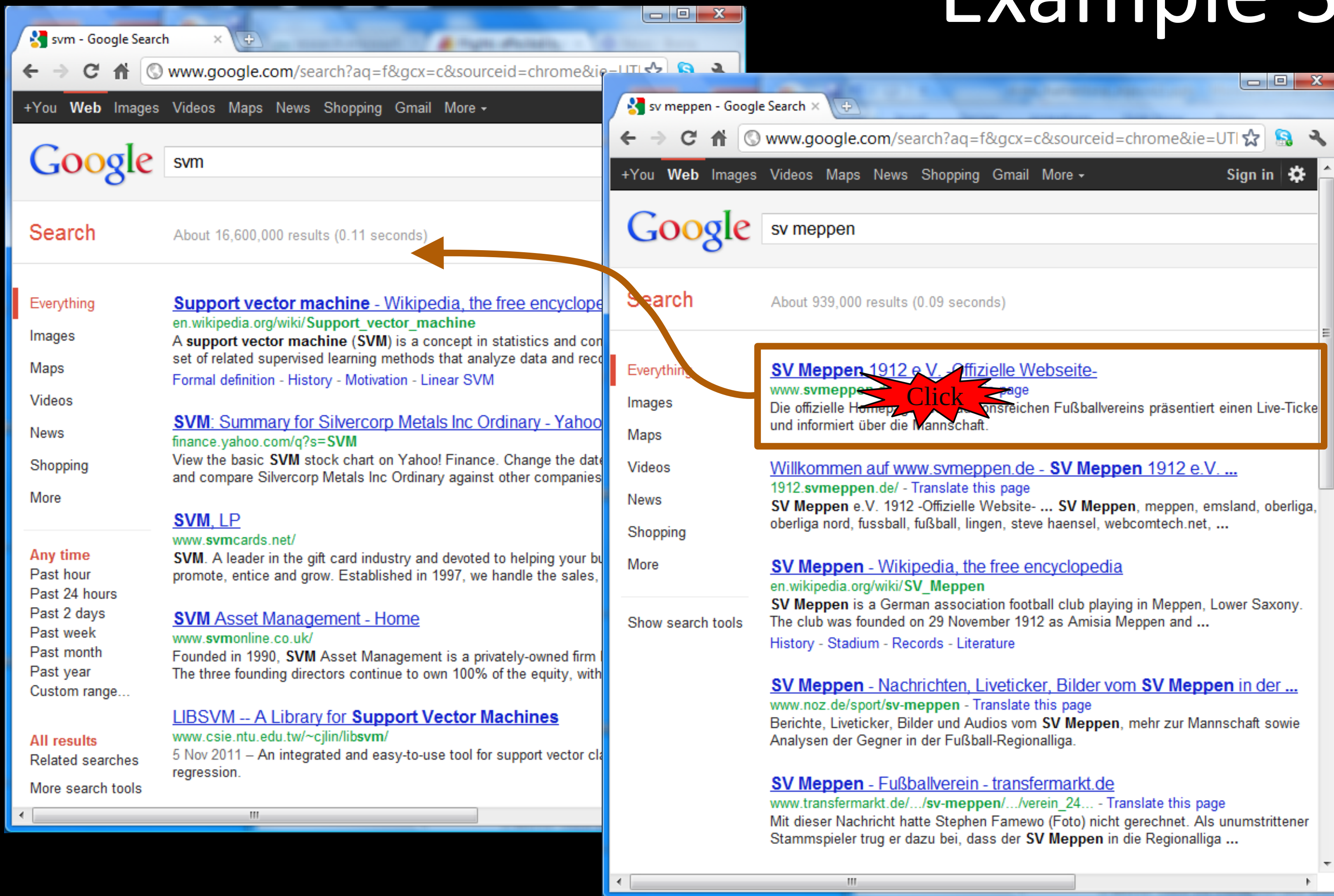
Who does the exploring?
Example 1



Who does the exploring?
Example 2

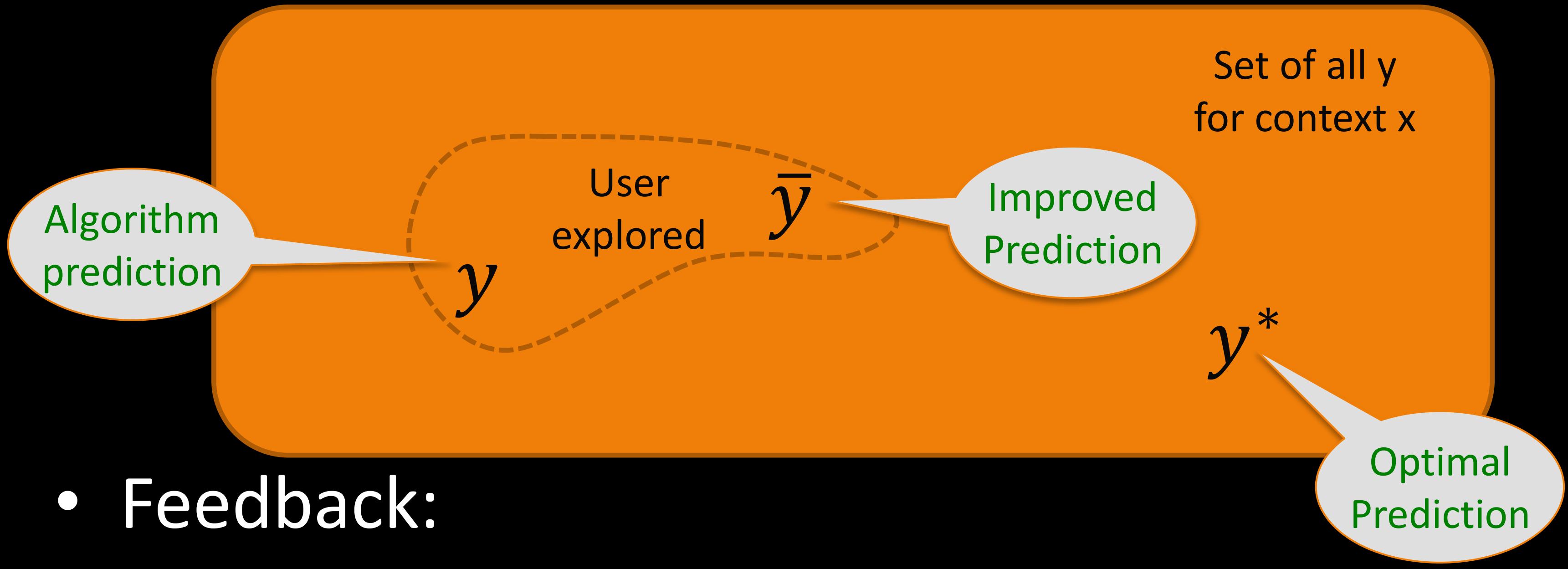


Who does the exploring? Example 3



Coactive Feedback Model

- Interaction: given x



- Feedback:
  - Improved prediction $\bar{y}_t$

$$U(\bar{y}_t|x_t) > U(y_t|x_t)$$

  - Supervised learning: optimal prediction y_t^*

$$y_t^* = \mathrm{argmax}_y\, U(y|x_t)$$

Machine Translation

x_t

We propose Coactive Learning as a model of interaction between a learning system and a human user, where both have the common goal of providing results of maximum utility to the user.

y_t

Wir schlagen vor, koaktive Learning als ein Modell der Wechselwirkung zwischen einem Lernsystem und menschlichen Benutzer, wobei sowohl die gemeinsame Ziel, die Ergebnisse der maximalen Nutzen für den Benutzer.

$\prec$

$\bar{y}_t$

Wir schlagen ~~vor,~~ koaktive Learning als ein Modell ~~der Wechselwirkung~~ des Dialogs zwischen einem Lernsystem und menschlichen Benutzer, wobei ~~sowohl die~~ beide das gemeinsame Ziel haben, die Ergebnisse der maximalen Nutzen für den Benutzer zu liefern.

Supervised Batch Learning

- Data: $(x, y^*) \sim P(X, Y^*)$
  - x is input
  - y^* is true label
- Rules: $f \in H$
- Prediction: $\hat{y} = f(x)$
- Loss function: $\Delta(\hat{y}, y^*)$

→ Find $f \in H$ that minimizes prediction error

$$R(f) = \int \Delta(f(x), y^*)$$

Coactive Preference Perceptron

- Model
  - Linear model of user utility: $U(y|x) = w^T \phi(x,y)$
- Algorithm
  - FOR t = 1 TO T DO
    - Observe x_t
    - Present $y_t = \text{argmax}_y \{ w_t^T \phi(x_t,y) \}$
    - Obtain feedback $\bar{y}_t$ from user
    - Update $w_{t+1} = w_t + \phi(x_t,\bar{y}_t) - \phi(x_t,y_t)$
- This may look similar to a multi-class Perceptron, but
  - Feedback $\bar{y}_t$ is different (not get the correct class label)
  - Regret is different (misclassifications vs. utility difference)

$$\mathrm{R(A)} = \frac{1}{T}\sum_{t=1}^{T}[U(y_t^*|x) - U(y_t|x)]$$

Never revealed:
- cardinal feedback
- optimal y^*

[Shivaswamy, Joachims, 2012]

Coactive Perceptron: Regret Bound

- Model

  $U(\mathbf{y}|\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x},\mathbf{y})$, where w is unknown

- Feedback: ξ-Approximately α-Informative

$$E[U(x_t, \bar{y}_t)] \geq U(x_t, y_t) + \alpha\big(U(x_t, y_t^*) - U(x_t, y_t)\big) - \xi_t$$

user feedback

system prediction

gap to optimal

model error

- Theorem

  For user feedback $\bar{\mathbf{y}}$ that is α-informative in expectation, the expected average regret of the Preference Perceptron is bounded by

$$E\left[\frac{1}{T}\sum_{t=1}^{T} U(y_t^*|x) - U(y_t|x)\right] \leq \frac{1}{\alpha T}\sum_{t=1}^{T}\xi_t + \frac{2R||w||}{\alpha\sqrt{T}}$$

→ zero

model error

[Shivaswamy, Joachims, 2012]

Preference Perceptron: Experiment

Experiment:

- Automatically optimize Arxiv.org Fulltext Search

Model

- Utility of ranking y for query x: $U_t(y|x) = \sum_i \gamma_i \, w_t^T \phi(x,y^{(i)})$ [~1000 features]

  → Computing argmax ranking: sort by $w_t^T \phi(x,y^{(i)})$

Analogous to DCG

Feedback

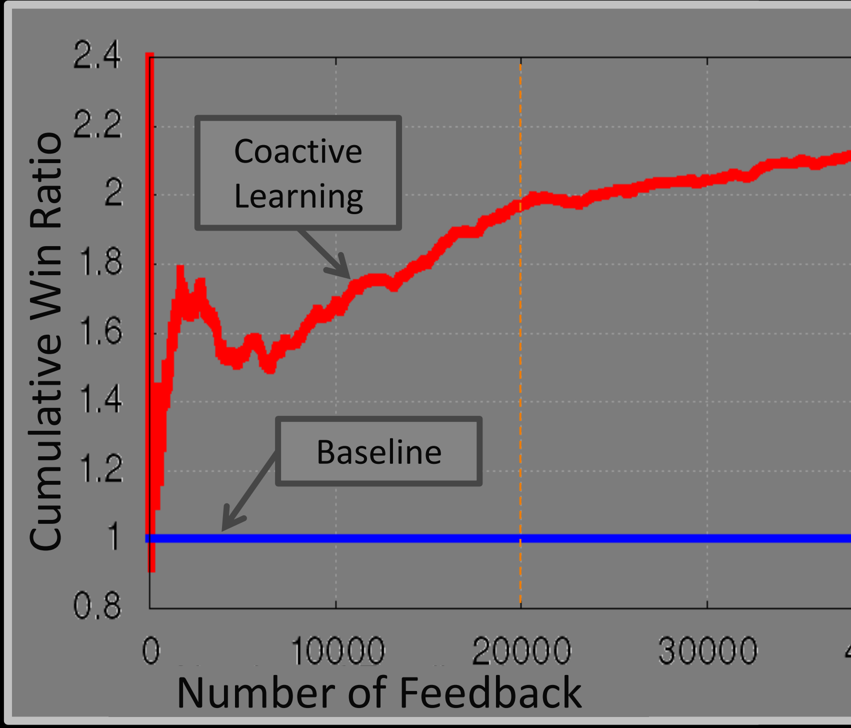
- Construct $\bar{y}_t$ from y_t by moving clicked links one position higher.
- Perturbation [Raman et al., 2013]

Baseline

- Handtuned w_{base} for $U_{base}(y|x)$

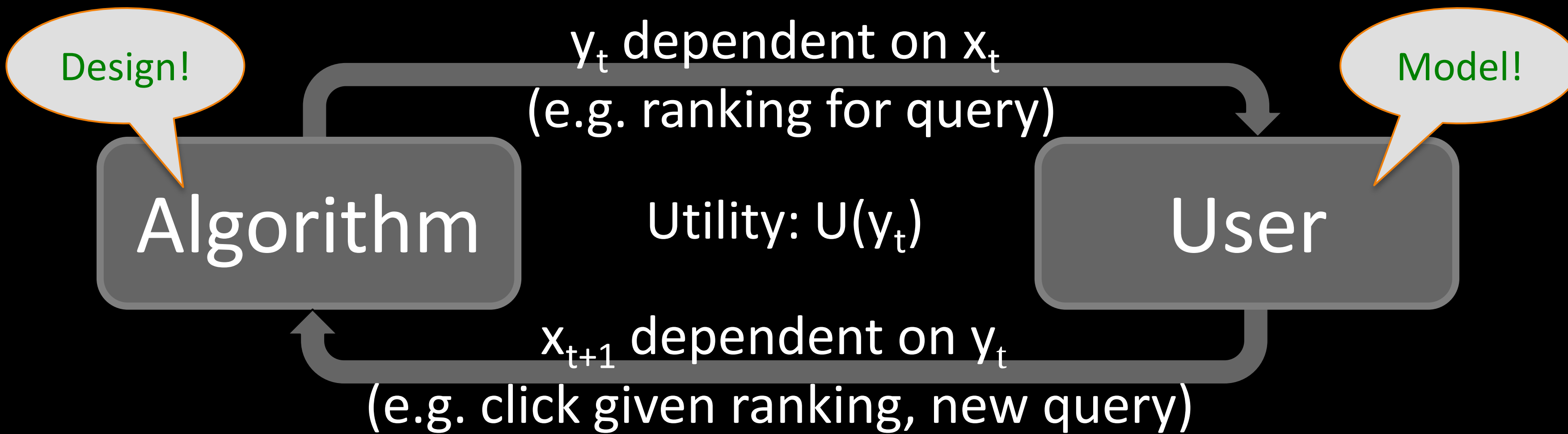
Evaluation

- Interleaving of ranking from $U_t(y|x)$ and $U_{base}(y|x)$



[Raman et al., 2013]

Interactive Learning System



- Observed Data ≠ Training Data
- Decisions → Feedback → Learning Algorithm
  - Dueling Bandits
    - → Model: Pairwise comparison test $P(y_i \succ y_j \mid U(y_i)>U(y_j))$
    - → Algorithm: Interleaved Filter 2, $O(|Y|\log(T))$ regret
  - Coactive Learning
    - → Model: for given y, user provides $\bar{y}$ with $U(\bar{y}|x) > U(y|x)$
    - → Algorithm: Preference Perceptron, $O(\|w\|\, T^{0.5})$ regret

Running
Interactive Learning Experiments

1) Build your own system and provide service
   → not your thing
   → too little data
2) Convince others to run your experiments on commercial system
   → good luck with that
3) Use large-scale historical log data from commercial system

Information in Interaction Logs

- Partial Information (aka "Bandit") Feedback
  - News recommender f_0 presents set y of articles for user x and observes that user reads δ minutes
  - Ad system f_0 presents ad y to user x and observes monetary payoff δ
  - Search engine f_0 interleaves ranking y for query x with baseline ranker and observes win/loss δ
  - MT system f_0 predicts translation y for x and receives rating δ

→ Data: $S = ((x_1, y_1, \delta_1), \ldots, (x_n, y_n, \delta_n))$

context

f_0 action

reward

Supervised Batch Learning

- Data: $(x, y^*) \sim P(X, Y^*)$
  - x is input
  - y^* is true label
- Rules: $f \in H$
- Prediction: $\hat{y} = f(x)$
- Loss function: $\Delta(\hat{y}, y^*)$

→ Find $f \in H$ that minimizes prediction error

$$R(f) = \int \Delta(f(x), y^*)$$

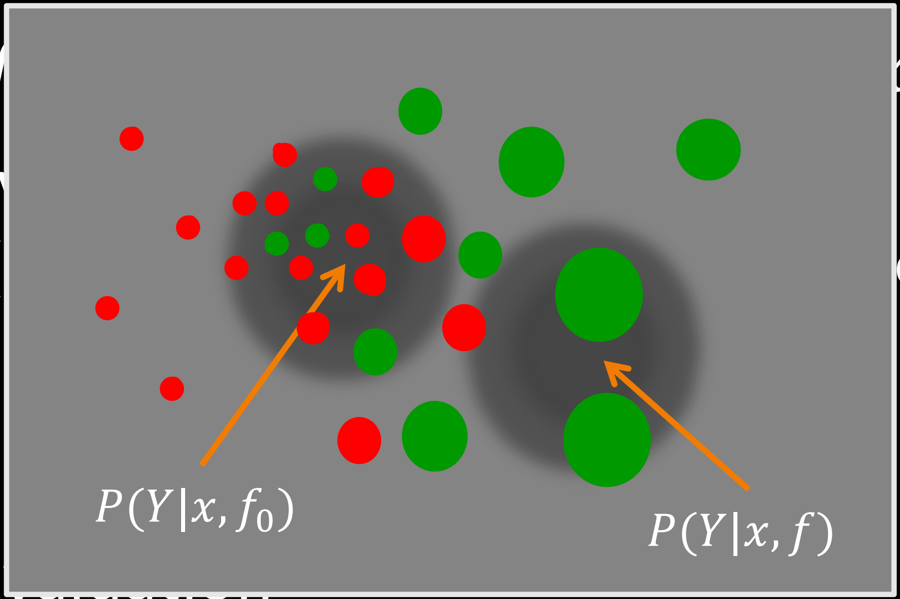
Changing History

- Expected Performance of Stochastic Policy f: $P(y|x, f)$

$$R(f$$

- On-Policy E

Given $S = ($ … ed under f_0,



- Off-Policy E

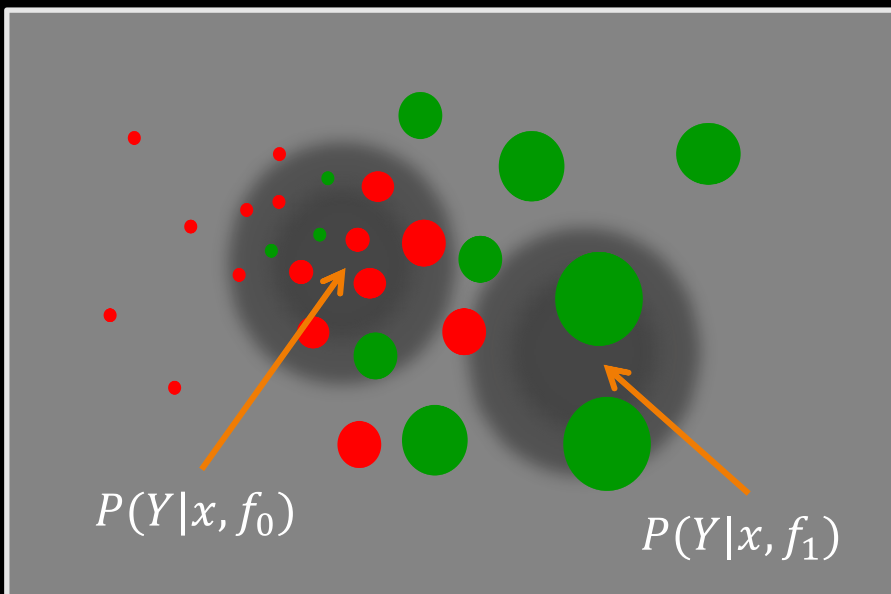
Given $S = ((x_1, y_1, \delta_1), \dots, (x_n, y_n, \delta_n))$ collected under f_0,

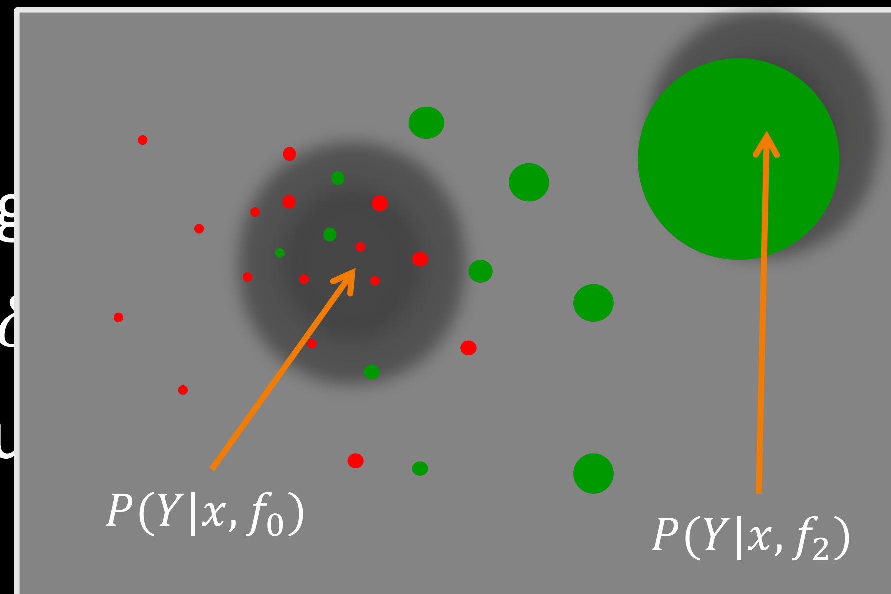
$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \delta_i \frac{P(y_i|x_i, f)}{P(y_i|x_i, f_0)}$$

Propensity weight

[Rubin, 1983] [Langford, Li, 2009.]

Partial Information Empirical Risk Minimization





$$\hat{f} := \operatorname{argmax}_{f \in H} \sum_{i}^{n} \frac{P(y_i | x_i, h)}{p_i} \delta_i$$

[Langford & Li], [Bottou, et al.,2014]

Partial Information Empirical Risk Minimization

- Setup
  - Stochastic logging using f_0 with $p_i = P(y_i | x_i, f_0)$

    → Data $S = \big((x_1, y_1, \delta_1, p_1), \dots, (x_n, y_n, \delta_n, p_n)\big)$
  - Stochastic prediction rules f ∈ H: $P(y_i | x_i, f)$
- Training

$$\hat{f} \coloneqq \mathrm{argmax}_{f \in H} \sum_{i}^{n} \frac{P(y_i | x_i, h)}{p_i} \, \delta_i$$

[Langford & Li], [Bottou, et al.,2014]

Propensity Risk Minimization

- Intuition and Learning Theory
  - De-bias estimator through propensity weighting
  - Correct for variance of estimators for different $f \in \mathrm{H}$
  - Account for capacity of the hypothesis space H
- Training: optimize generalization error bound

$$\hat{f} = \operatorname{argmax}_{f \in H} \left[\begin{array}{c} \widehat{Mean}\left(\frac{P(y_i|x_i,f)}{p_i}\delta_i\right) \\ -\lambda_1 \sqrt{\widehat{Var}\left(\frac{P(y_i|x_i,f)}{p_i}\delta_i\right)} \\ -\lambda_2\, Reg(H) \end{array} \right]$$

Unbiased Estimator

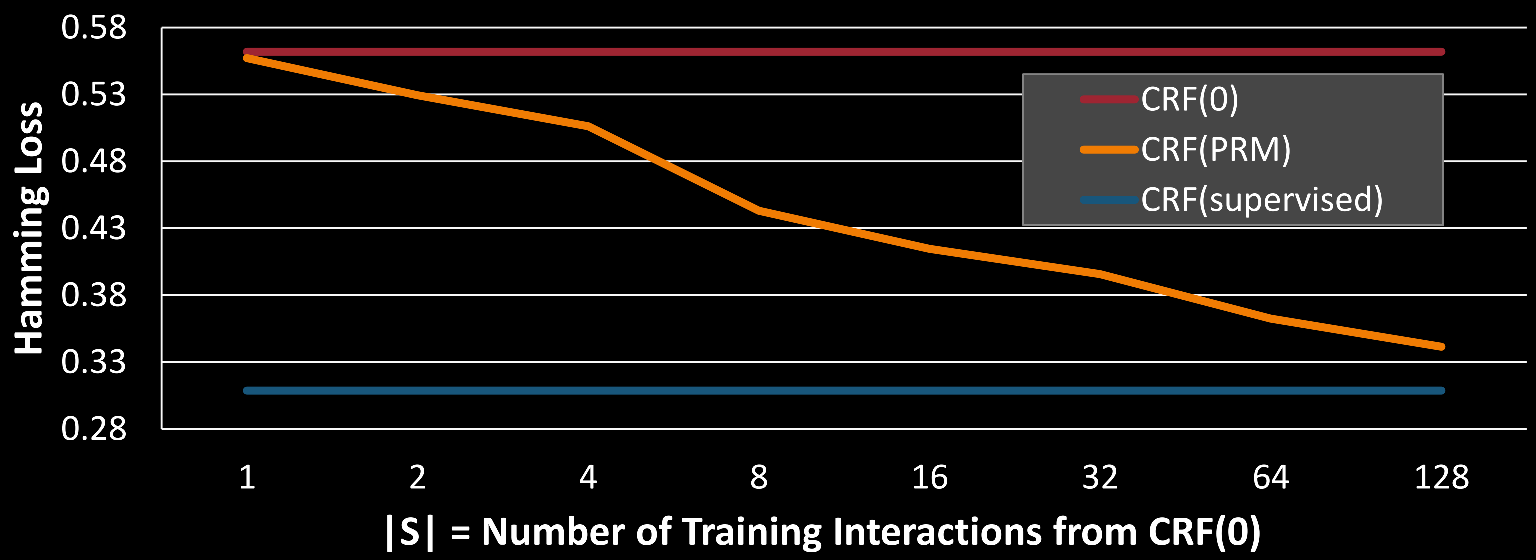
Variance Control

Capacity Control

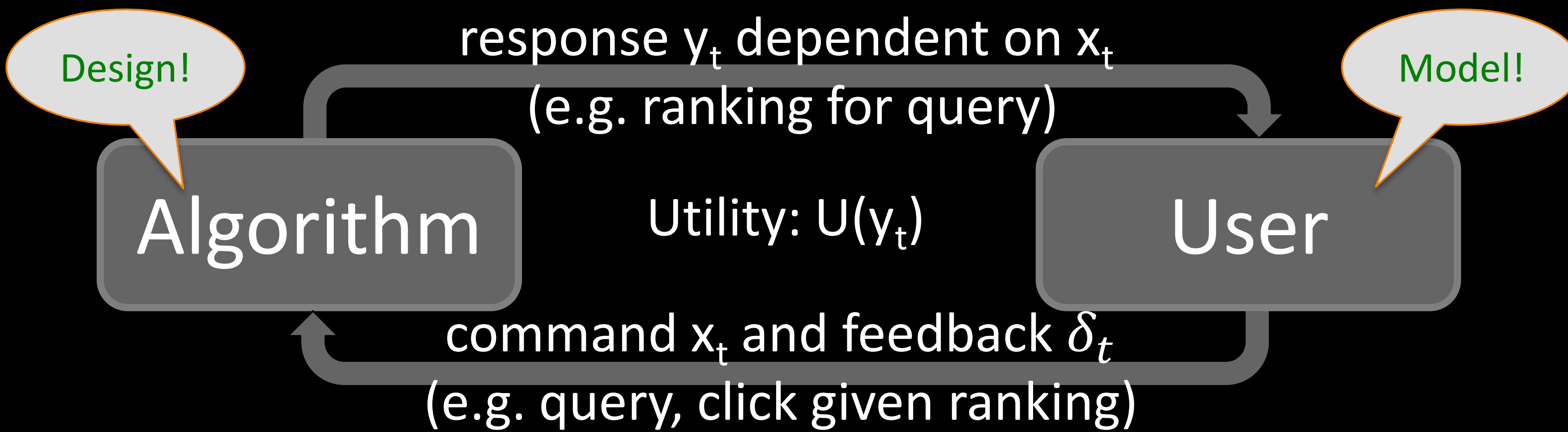
[Swaminathan, Joachims]

Experiment: Propensity Risk Minimization

- Experiment Setup
  - x: Reuters RCV1 text document
  - y: label vector with 4 binary labels
  - δ: number of correct labels
  - p: propensity under logging with CRF(0)
  - H: CRF with one weight vector per label
- Results

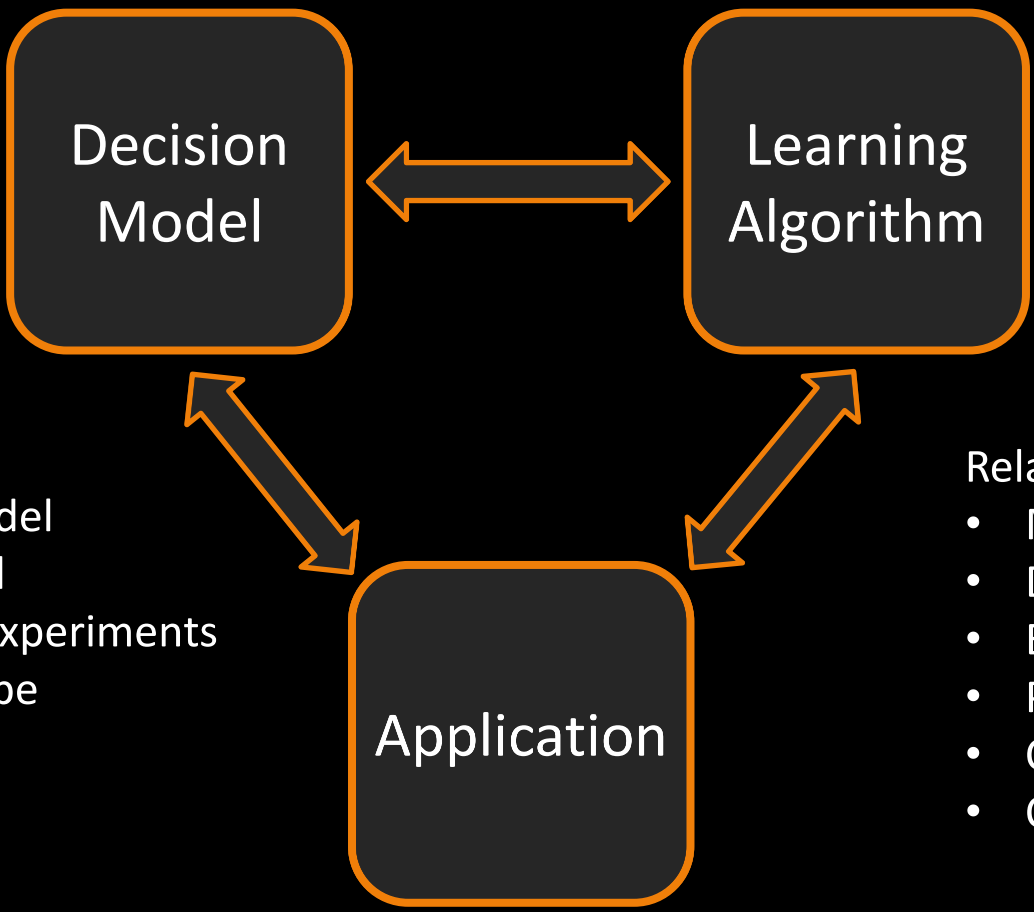


Summary and Conclusions



- Observed Data ≠ Training Data
- Decisions → Feedback → Learning Algorithm
  - Dueling Bandits
    - → Model: Pairwise comparison test $P(y_i \succ y_j \mid U(y_i)>U(y_j))$
    - → Algorithm: Interleaved Filter 2, $O(|Y|\log(T))$ regret
  - Coactive Learning
    - → Model: for given y, user provides $\bar{y}$ with $U(\bar{y}|x) > U(y|x)$
    - → Algorithm: Preference Perceptron, $O(\|w\|\ T^{0.5})$ regret Data
  - Propensity Risk Minimization
    - → Partial information learning in the Batch setting

Learning from Human Decisions



Design Space:

- Decision Model
- Utility Model
- Interaction Experiments
- Feedback Type
- Regret
- Applications

Related Fields:

- Micro Economics
- Decision Theory
- Econometrics
- Psychology
- Communications
- Cognitive Science

Contact: tj@cs.cornell.edu

Software + Papers: www.joachims.org