Machine Learning for Search Ranking and Ad Auctions

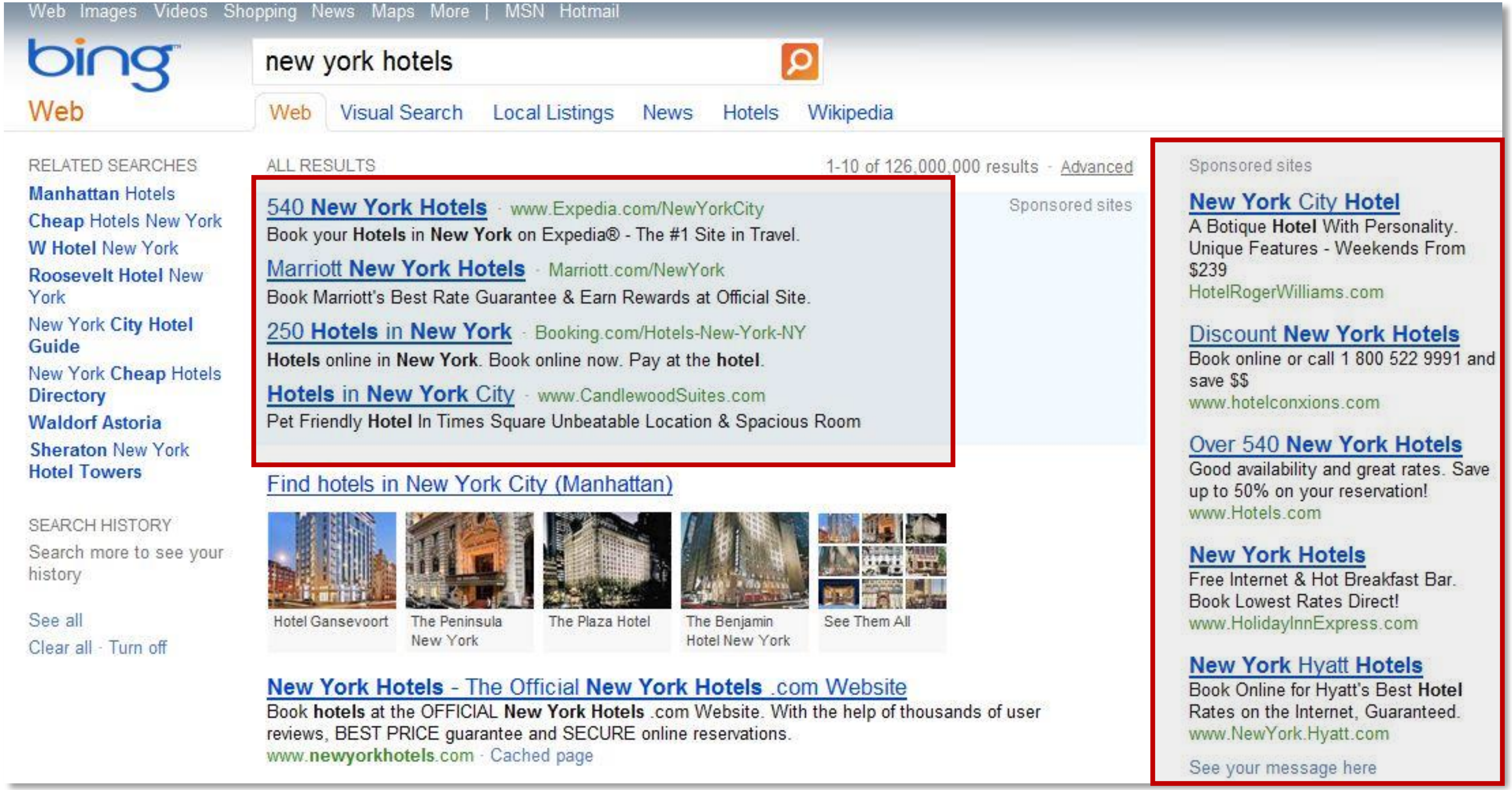
Tie-Yan Liu
Senior Researcher / Research Manager
Microsoft Research

The Speaker

- Research interests
  - Algorithms
    - Learning to rank
    - Knowledge-powered word embedding (deep learning)
    - Fast and scalable topic models
    - Game theoretic machine learning
    - Online learning
  - Theory
    - Statistical Learning theory for ranking, deep learning, and game-theoretic learning
  - Platform
    - Parallel machine learning

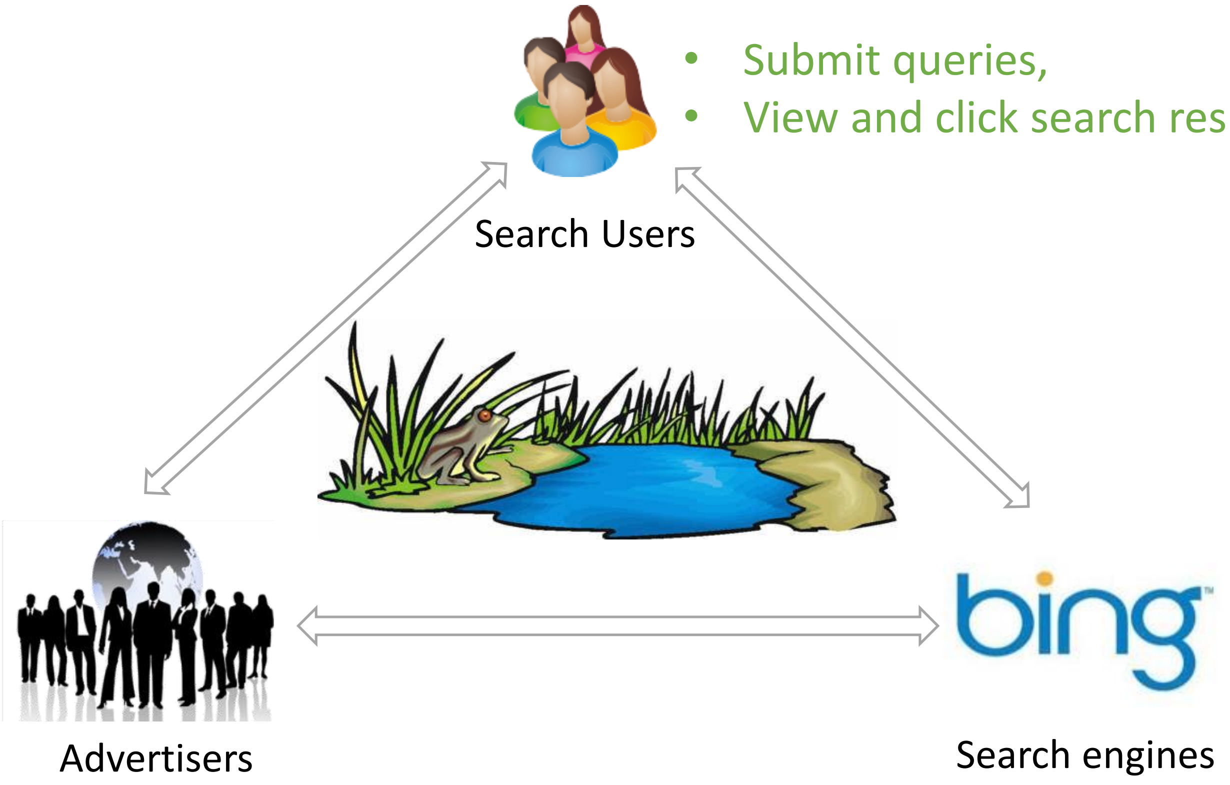
- Professional activities
  - Book @ Springer: Learning to rank for information retrieval
  - 70+ papers at top conferences and journals (with 8000+ citations), including 2 best paper awards
  - Chairs/keynote speakers of 10+ top conferences in machine learning, information retrieval, and computational economics
  - Associate editor of ACM TOIS, IR Journal, and FnTIR, etc.
  - Adjunct faculty of CMU (LTI), USTC, Nankai, and SYSU.
  - Honorary professor of University of Nottingham

Search and Ads



Eco-system

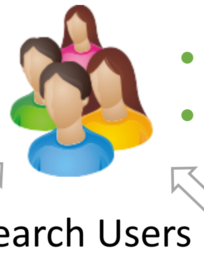
- Submit queries,
- View and click search results and ads



- Retrieve/rank web pages according to relevance to query
- Select ads and predict their click probabilities
- Run auction to determine ranking and pricing of ads
- Display search results and ads to users, charge advertiser when their ads are clicked

- Provide ad copies,
- Bid on keywords for ads,
- Pay if ads are clicked by users.

Eco-system

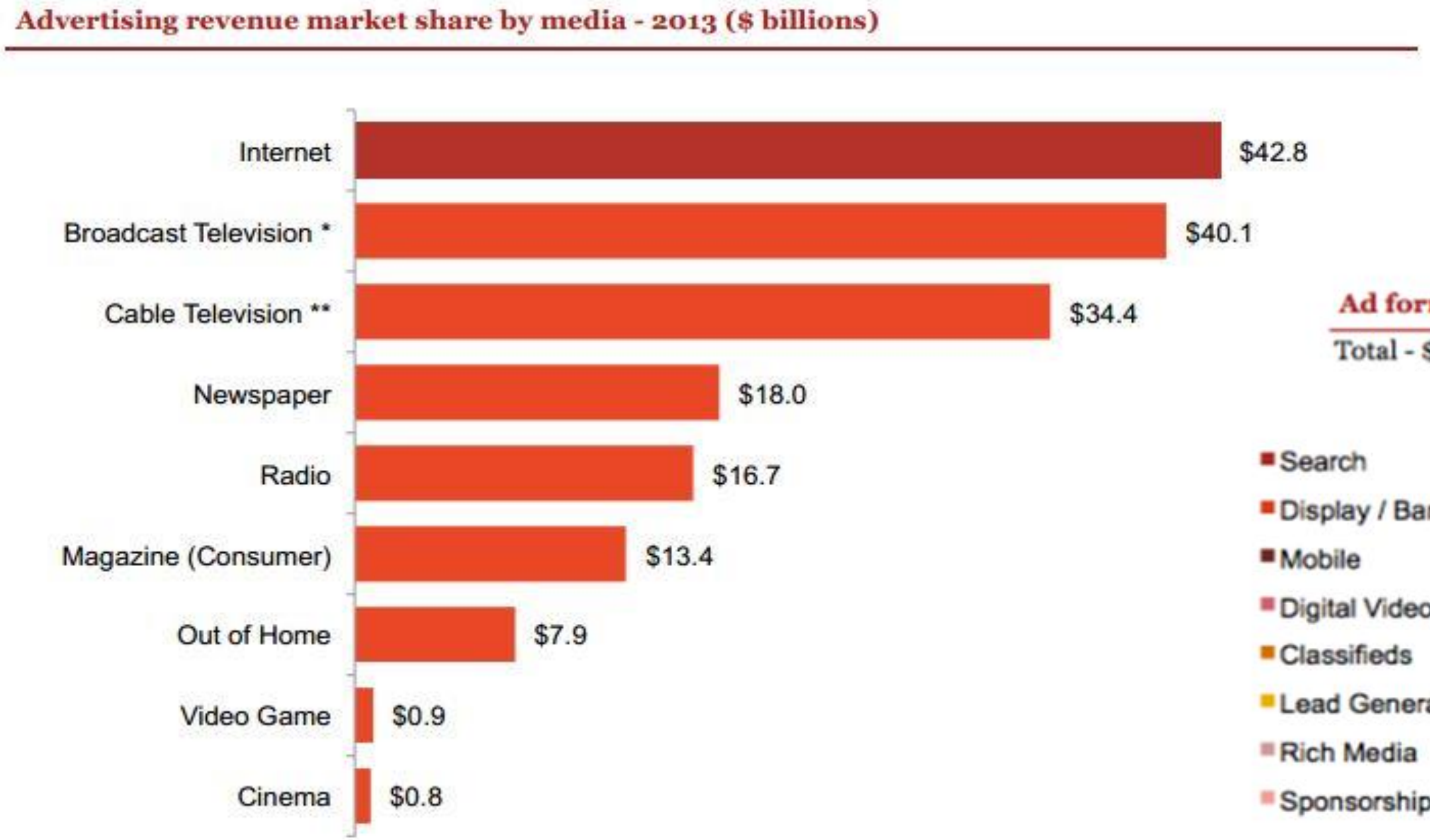


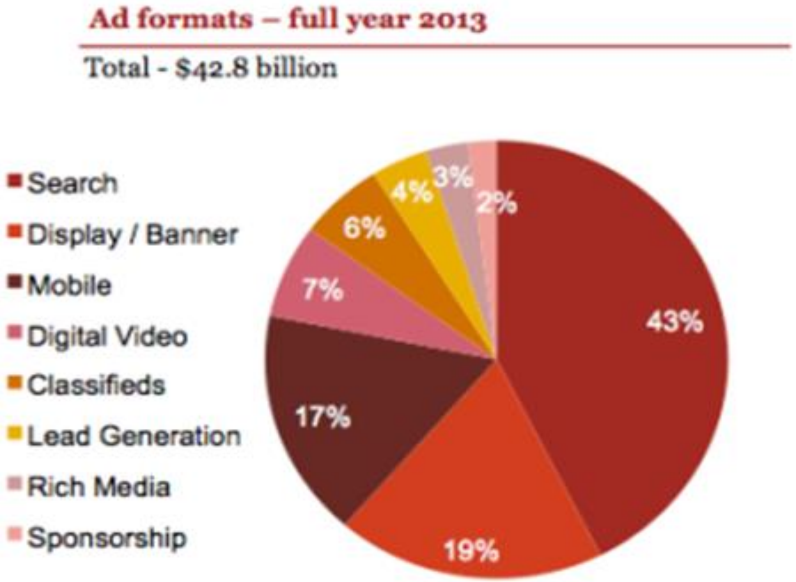
- Submit queries,
- View and click search results and ads

How much money do you contribute to the search engine every time you submit a query?

Revenue per search: 5~10 cents
Daily profit: ~10 Million dollars

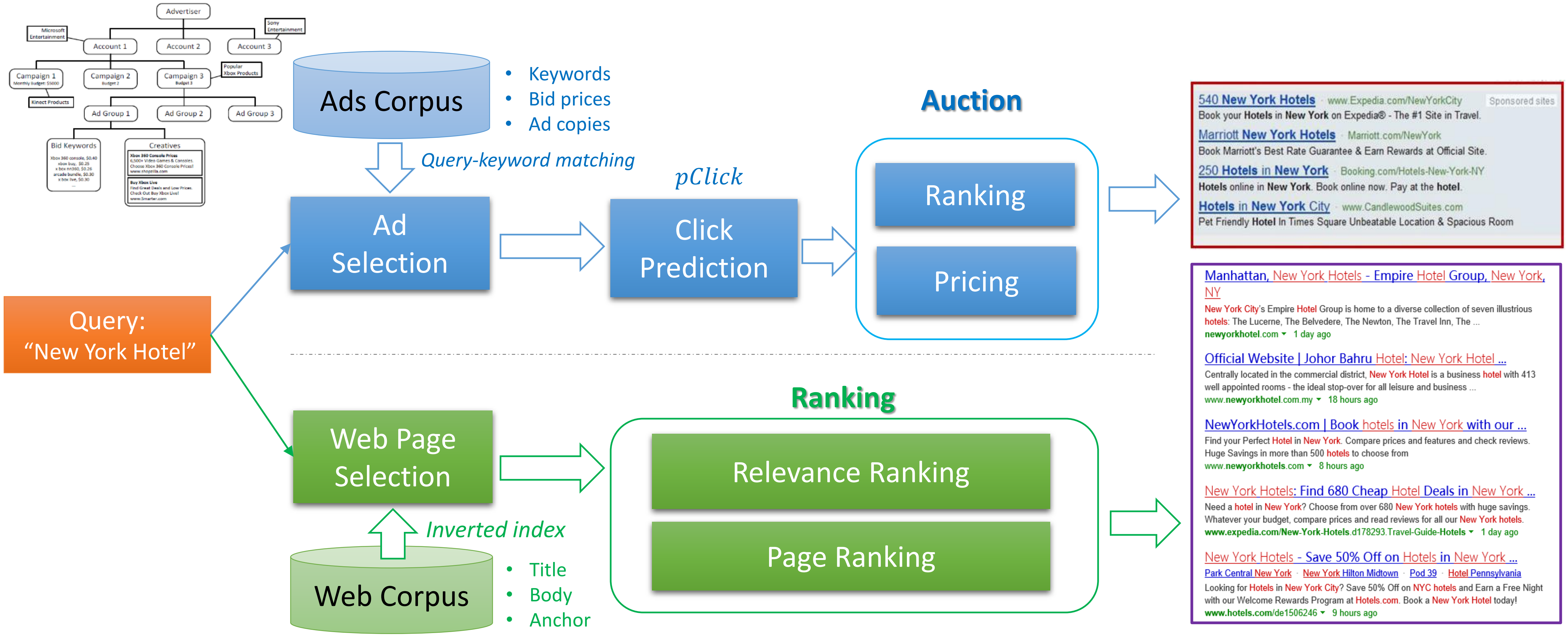
Cash Machine – $42.8 billion in 2013





Sources: IAB/PwC Internet Ad Revenue Report, 2013; PwC

Backend Systems



Key Components

Search – Ranking

- Heuristically designed ranking models
  - $f_{BM25}(D,Q) = \sum_i \frac{IDF(q_i) f(q_i,D)(k_1+1)}{f(q_i,D)+k1(1-b+b\frac{|D|}{AVDL})}$,
    $$IDF(q_i) = \log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$
  - $PRank(v_i) = \sum_{v_j \in inlink[v_i]} \frac{PRank(v_j)}{|outlink[v_j]|}$

Ads – Auction

- Heuristically designed ranking and pricing rules
  - GSP auction ranks ads by $pClick \times bid$
  - GSP auction charges clicked ads by $\frac{pClick_{next} \times bid_{next}}{pClick}$.
- Under rationality assumption
  - For GSP, there always exists at least one pure Nash equilibrium
  - The worse-case social welfare in equilibrium is around 80% of the optimal social welfare.

Key Components

Search – Ranking

- Heuristically designed ranking models
  - $f_{BM25}(D,Q) = \sum_i \frac{IDF(q_i) f(q_i,D)(k_1+1)}{f(q_i,D)+k1(1-b+b\frac{|D|}{AVDL})}$,

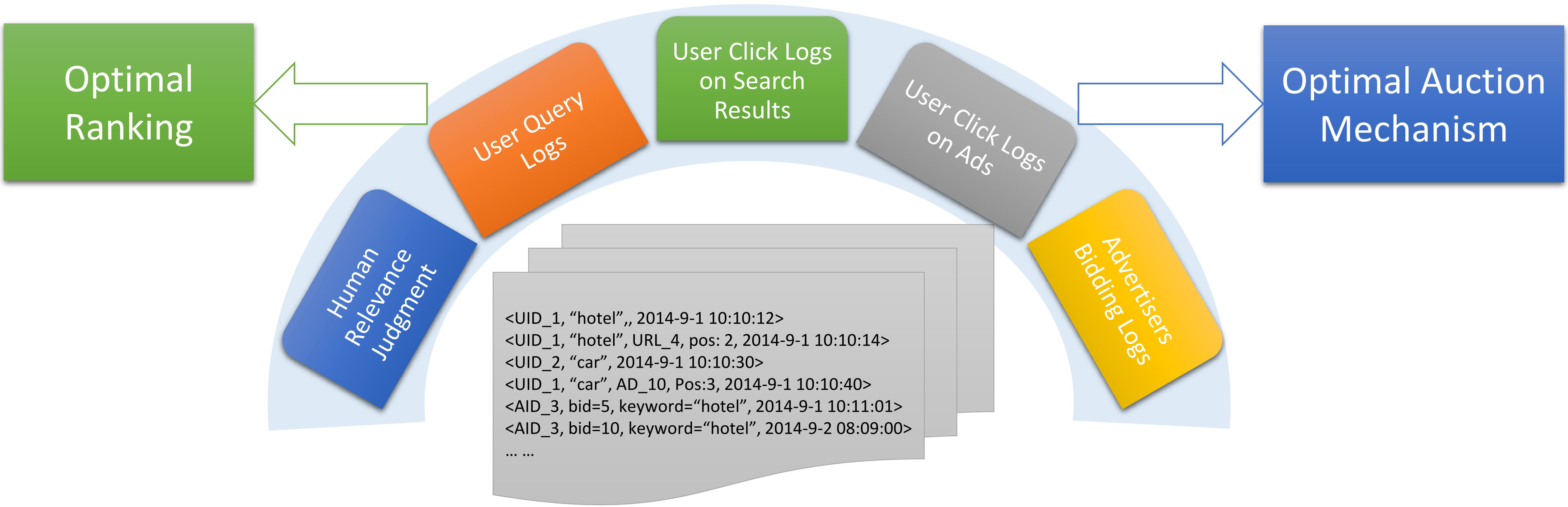
$IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{\ldots}$

Ads – Auction

- Heuristically designed ranking and pricing rules
  - GSP auction ranks ads by $pClick \times bid$
  - GSP auction charges clicked ads by $\frac{pClick_{next} \times bid_{next}}{pClick}$.

- Conventional approaches are based on heuristically designed magic formulas, or strong assumptions.
- Are these heuristic methods optimal? Can we improve them in an effective way?

“Data” vs. “Heuristics + Assumptions”



Instead of relying on heuristics and assumptions, we can let the data speak for us

Part I: Machine Learning for Search Ranking

Is Ranking a New Problem?

Pointwise approach

1
- **Reduce ranking to regression**
  - Treat relevance degree (click frequency) as real values
  - Example: Regression [Cossock and Zhang, 2006].

2
- **Reduce ranking to classification**
  - Treat relevance degree (or click event) as categories
  - Example: MC-Rank [Li, et al. 2007].

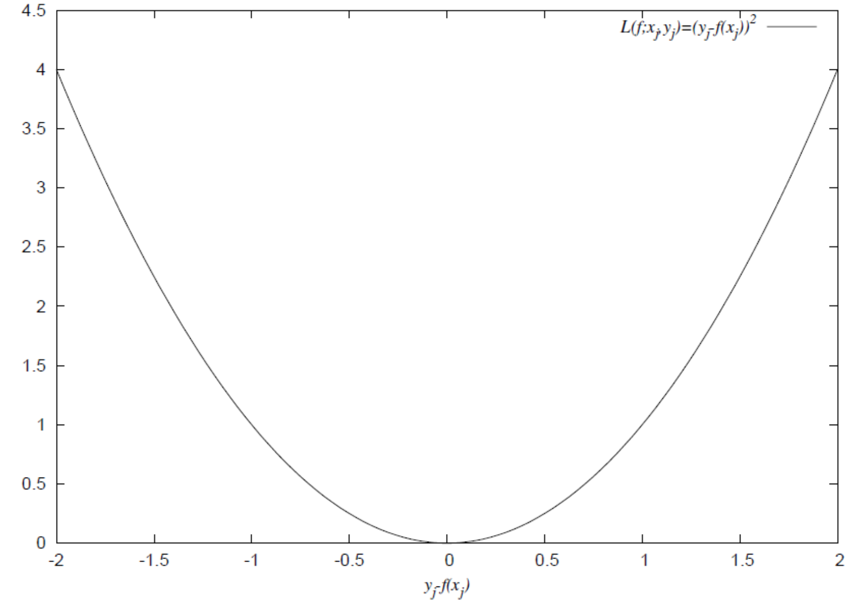
Pairwise approach

3
- **Reduce ranking to pairwise classification**
  - Classify the order between each pair of documents.
  - Example: RankSVM [Herbrich, et al. 2000], RankBoost [Freund, et al. 2003].

Example: Subset Ranking

[Cossock and Zhang, COLT 2006]

- Regard relevance degree as real number, and use regression to learn the ranking function.
  - $L(f; x_j, y_j) = \left(f(x_j) - y_j\right)^2$



Regression-based

Example: McRank

[Li, et al. NIPS 2007]

- Multi-class classification is used to learn the ranking function.
  - For document x_j, the output of the classifier is $\hat{y}_j$.
  - Loss function: surrogate function of $I_{\{y_j \neq \hat{y}_j\}}$
- Ranking is produced by combining the outputs of the classifiers.

$$\hat{p}_{j,k} = P(\hat{y}_j = k), \quad f(x_j) = \sum_{k=1}^{K} \hat{p}_{j,k} k$$

Classification-based

Example: Ranking SVM

[Herbrich, et al. , Advances in Large Margin Classifiers, 2000]

- Ranking SVM is rooted in the framework of SVM
- Kernel tricks can also be applied to Ranking SVM, to handle complex non-linear problems.

$$\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \sum_{u,v:y_{u,v}^{(i)}=1} \xi_{u,v}^{(i)}$$

$$w^T \left(\boxed{x_u^{(i)} - x_v^{(i)}} \right) \geq 1 - \xi_{u,v}^{(i)}, \text{if } y_{u,v}^{(i)} = 1.$$

$$\xi_{uv}^{(i)} \geq 0, i = 1, \ldots, n.$$

$x_u - x_v$ as positive instance of learning

Use SVM to perform binary classification on these instances, to learn model parameter *w*

Pairwise Classification-based

Appropriate Reductions?

- "Rank" as learning target
  - Top-ranked documents are more important
  - Relative order > absolute score
  - Evaluations are rank-based (NDCG, MAP, etc.)
    - $AP = \frac{\sum_k P@k \cdot g(\pi^{-1}(k))}{\#\{\text{docs with ground−truth label 1}\}}$,
    - $NDCG@K = Z_k \sum_k G(\pi^{-1}(k))/\log(k+1)$
- "Query" as important notion
  - Documents are comparable only w.r.t. the same query
  - Evaluations are averaged over queries

Something important for ranking is missing…

Appropriate Reductions?

- Example:
  - Model f : $f(A) = 3, f(B) = 0, f(C) = 1$ ACB
  - Model h : $h(A) = 4, h(B) = 6, h(C) = 3$ BAC
  - ground truth g: $g(A) = 6, g(B) = 4, g(C) = 3$ ABC
  - According to NDCG or AP: $sim(f,g) > sim(g,h)$
- Pointwise distance/pairwise comparison contradicts ranking measures
  - According to pointwise distance: $sim(f,g) < sim(g,h)$.
  - According to pairwise comparison: $sim(f,g) = sim(g,h)$.

Listwise Approach

- Instead of considering individual documents or document pairs, treat the entire document set $x^q = (x_1^q, \ldots, x_M^q)$ associated with the same query as a learning instance.
- Define "listwise" loss function on ranked list (permutation) of these documents
- Learn the ranking model by minimizing the listwise loss function

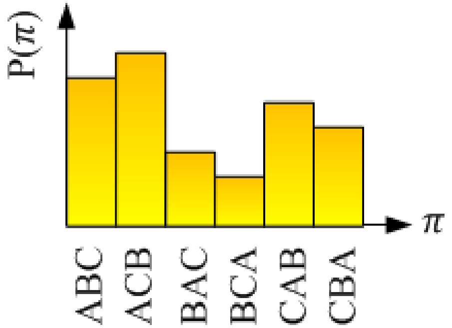
Notion of query is naturally captured.
Ranking positions are visible to the learning algorithm.

Representation of Ranked Lists

- Ranked list $\leftrightarrow$ Permutation probability distribution

f: $f(\mathrm{A})=3, f(\mathrm{B})=0, f(\mathrm{C})=1$;
Ranking by f: ABC

$\leftrightarrow$ $P(\pi \mid f)$



Plackett-Luce Model

$$P_{PL}(\pi \mid f) = \prod_{j=1}^{m} \frac{\exp(f(x_{\pi(j)}))}{\sum_{k=j}^{m} \exp(f(x_{\pi(k)}))}$$

$$P_{PL}(\mathrm{ABC} \mid f) = \boxed{\frac{\exp(f(\mathrm{A}))}{\exp(f(\mathrm{A})) + \exp(f(\mathrm{B})) + \exp(f(\mathrm{A}))}} \cdot \boxed{\frac{\exp(f(\mathrm{B}))}{\exp(f(\mathrm{B})) + \exp(f(\mathrm{C}))}} \cdot \boxed{\frac{\exp(f(\mathrm{C}))}{\exp(f(\mathrm{C}))}}$$

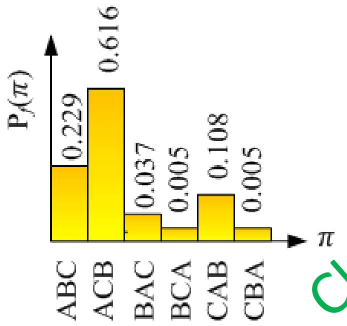
P(A ranked No.1)

P(B ranked No.2 | A ranked No.1)

P(C ranked No.3 | A ranked No.1, B ranked No.2)

Distance between Ranked Lists

f: $f(\mathrm{A}) = 3, f(\mathrm{B})=0, f(\mathrm{C})=1$;
Ranking by f: ABC $\leftrightarrow$



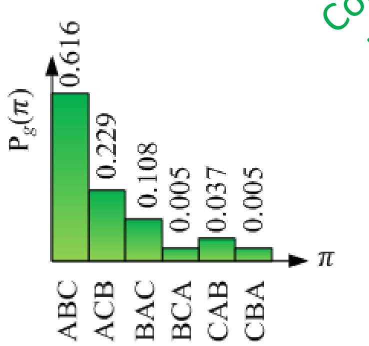
*Using **KL-divergence** to measure difference between distributions*

Closer!

Consistent with NDCG and AP

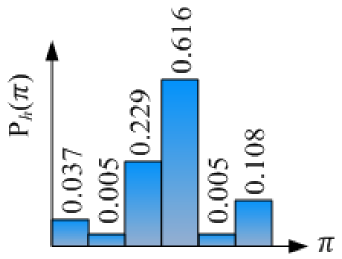
$$d(f,g) = 0.46$$

g: $g(\mathrm{A}) = 6, g(\mathrm{B})=4, g(\mathrm{C})=3$;
Ranking by g: ABC $\leftrightarrow$



$$d(g,h) = 2.56$$

h: $h(\mathrm{A}) = 4, h(\mathrm{B})=6, h(\mathrm{C})=3$;
Ranking by h: ACB $\leftrightarrow$

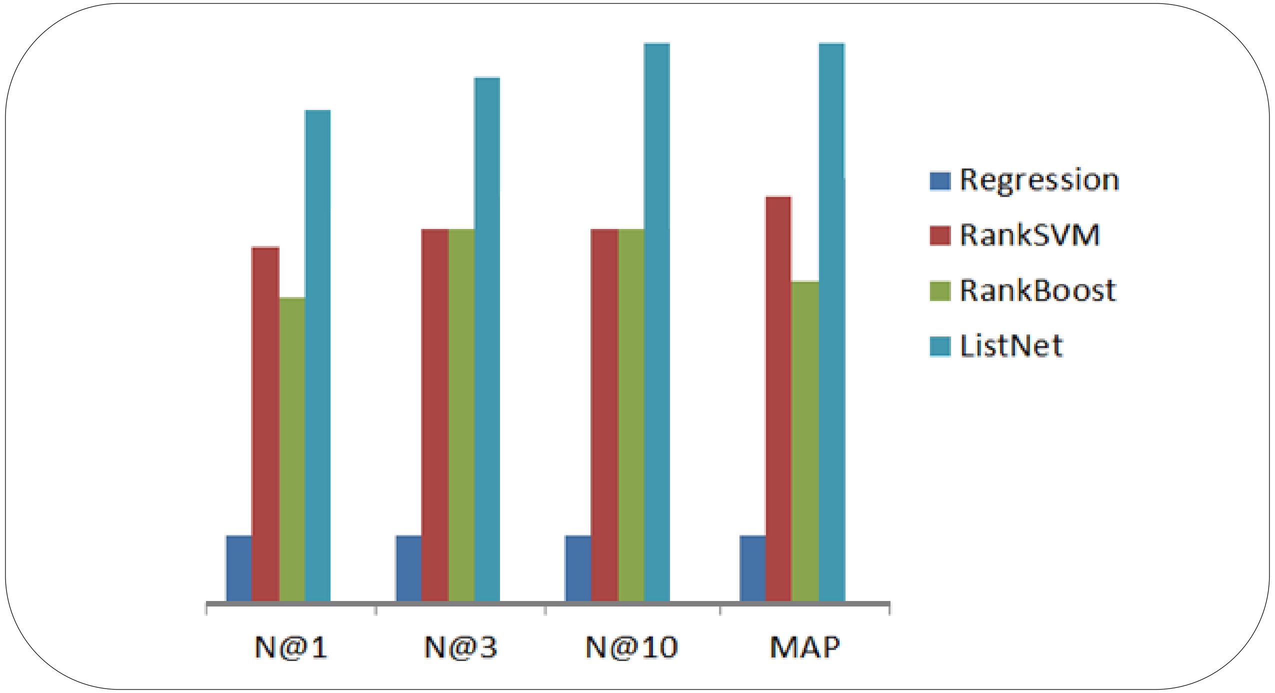


Listwise Loss Functions

- ListNet [Cao et al. ICML 2007]
  - Minimize KL divergence between permutation probabilities of ranking model and ground truth
  - $L(f;\mathbf{x},g) = D(P_{PL}(\pi|g)||P_{PL}(\pi|f(\mathbf{x})))$
- ListMLE [Xiao et al. ICML 2008]
  - Directly maximize likelihood of ground truth permutation induced by ranking model
  - $P_{PL}(\pi|g) \rightarrow P_g(\pi) = \begin{cases} 1, \pi = \pi_g \\ 0, \text{otherwise} \end{cases} \rightarrow L(f;\mathbf{x},g) = -\log P_{PL}(\pi_g|f(\mathbf{x}))$
- Model general ground truth labels using "*equivalent permutation set*" [Liu, FnTIR 2009]
  - $L(f;\mathbf{x},g) = \min_{\pi \in \Omega_g} (-\log P_{PL}(\pi|f(\mathbf{x})))$
  - For relevance degree or clicks: $\Omega_g = \{\pi|u < v, if\ g(\pi^{-1}(u)) > (\pi^{-1}(v))\}$
  - For pairwise preference: $\Omega_g = \{\pi|u < v, if\ g\left(\pi^{-1}(u), \pi^{-1}(v)\right) = 1\}$

Experimental Results on LETOR Benchmark

Winning number: comparison with other algorithms over all 7 sub datasets in LETPOR

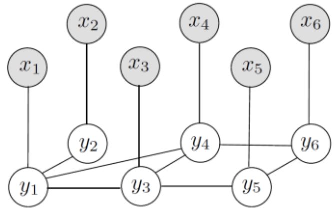


{ListNet}>{RankSVM, RankBoost}>{Regression}

http://research.microsoft.com/~letor/

Listwise Ranking Functions

- Beyond pointiwse ranking function
  - Instead of simply sorting the scores assigned to individual documents, also consider relationship between documents (topic diversity, domain hierarchy).
- C-CRF [Qin et al. NIPS 2008]

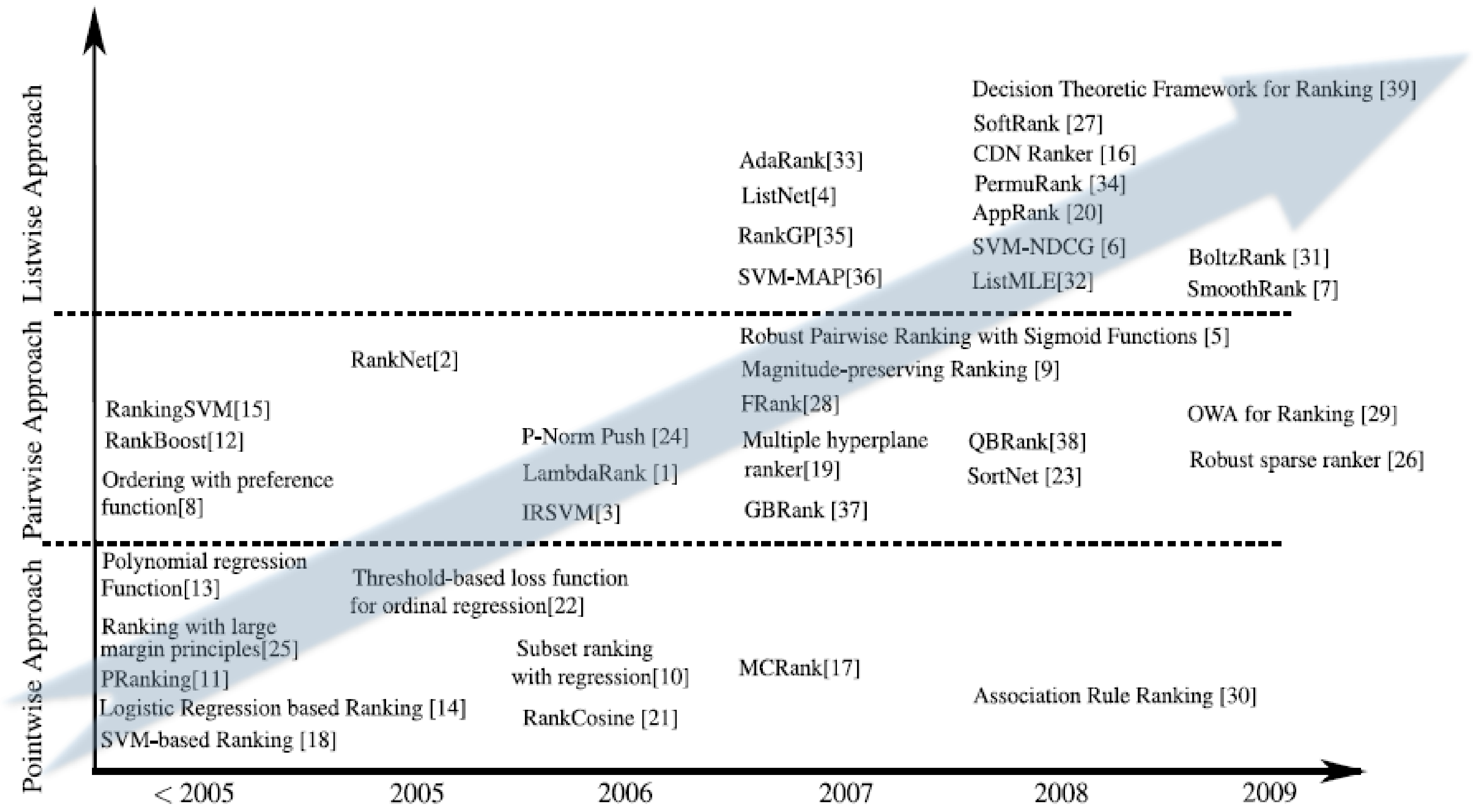


$$P(g^q|\mathbf{x}^q) = \frac{1}{Z(\mathbf{x}^q)} \exp\{\sum_i \sum_k \alpha_k h_k^1(g_i^q, \mathbf{x}^q) + \sum_{i,j} \sum_k \beta_k h_k^2(g_i^q, g_j^q, \mathbf{x}^q)\}\}$$

- R-RSVM [Qin et al. WWW 2008]

$$f(x) = \arg\min_z\{\alpha l_1(h(x;w), z) + \beta l_2(R, z)\}$$

Listwise Ranking: An Important Branch



Algorithmic development of learning to rank is booming

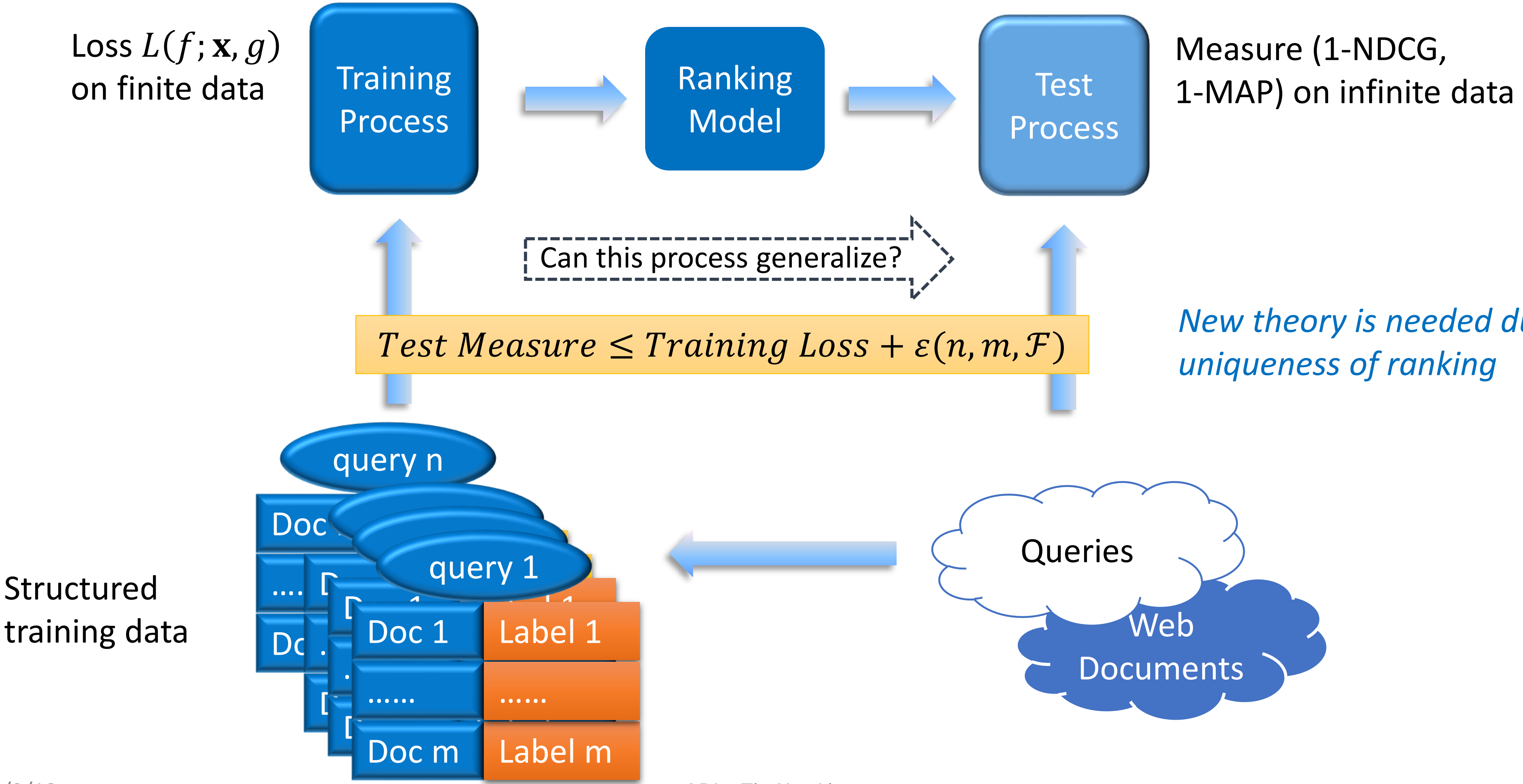
Beyond Empirical Results

- In practice, one can only observe experimental results on relatively small datasets. Such empirical results might not be reliable, because
  - Small training set cannot fully realize the potential of a learning algorithm.
  - Small test set cannot reflect the true performance of an algorithm, since the real query space is too huge.
- Statistical learning theory analyzes the performance of an algorithm when the training data is infinite and the test data is randomly sampled.

Generalization Analysis

- In the training phase, one learns a model by minimizing the empirical risk $\hat{R}(f)$ on the training data.
- In the test phase, one evaluates the expected risk $R(f)$ of the model on any sample.
- Generalization analysis is concerned with the asymptotical bound of the difference between the expected and empirical risks, when the number of training data approaches infinity.

Generalization Analysis for Learning to Rank



New theory is needed due to uniqueness of ranking

How to Get There…

$$Test\ Measure \leq Training\ Loss + \varepsilon(n, m, \mathcal{F})$$

$$\text{Test Measure} \leq \underbrace{\text{Test Loss} \leq \text{Training Loss} + \varepsilon}_{1}$$

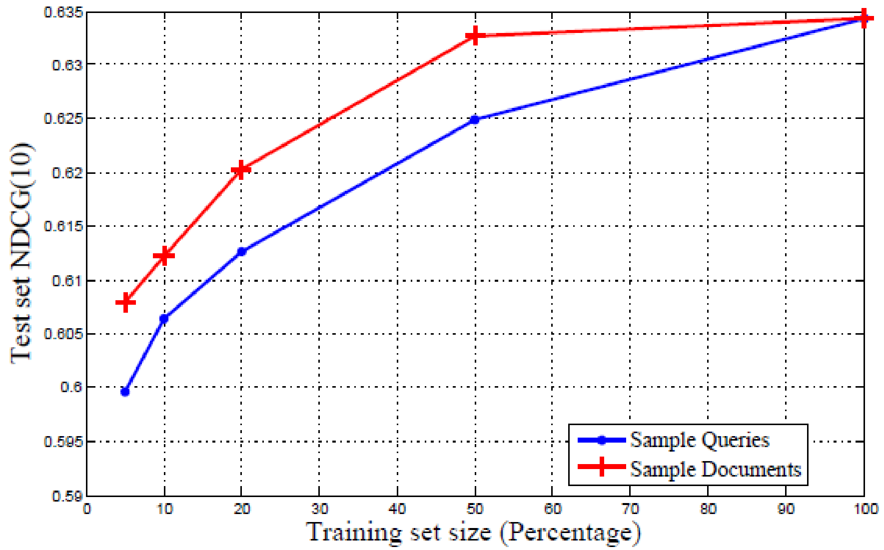
To perform this generalization analysis, we need to make probabilistic assumptions on the data generation.

Previous Assumptions

Instance Ranking [Agarwal et al., 2005; Clemencon et al.,2007]

No Notion of query

"Deep and shallow training sets correspond to the same generalization ability".

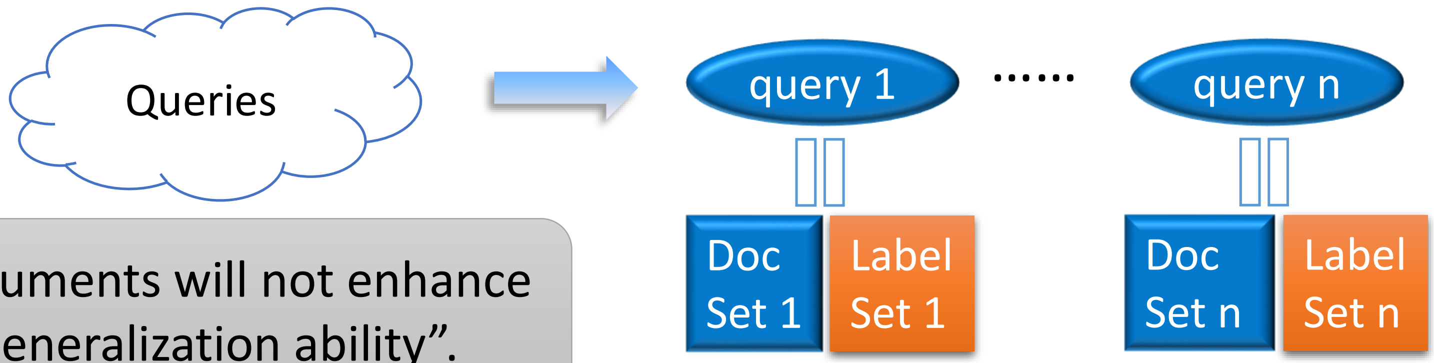


[Yilmaz and Robertson, 2009]

Previous Assumptions

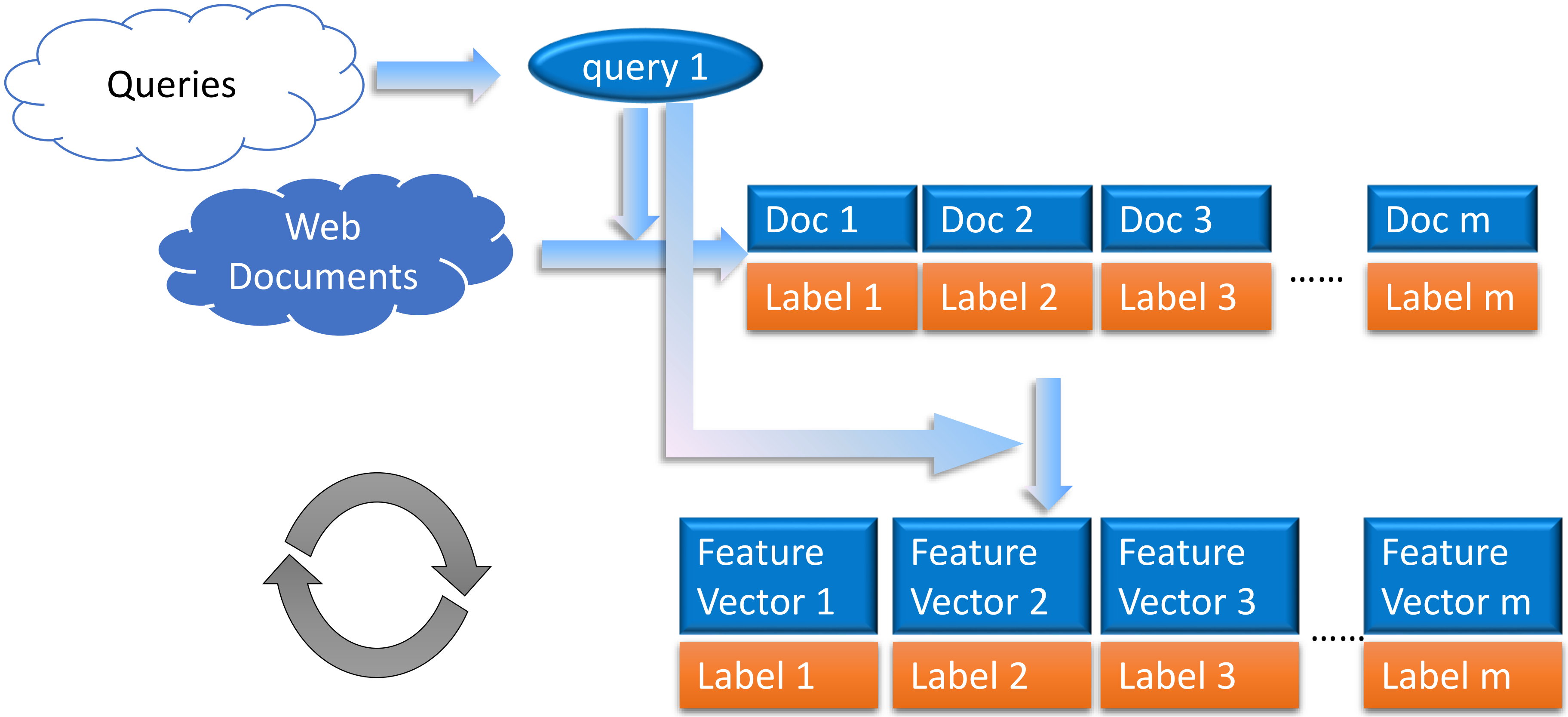
Instance Ranking [Agarwal et al., 2005; Clemencon et al.,2007]

Subset Ranking [Lan et al.,2008; Lan et al.,2009]



"More training documents will not enhance and even hurt generalization ability".

Two-layer Sampling
[Chen et al. NIPS 2010]



Two-layer Sampling

[Chen et al. NIPS 2010]

- Different from instance ranking
  - Sampling of queries
  - Documents associated with different queries are sampled according to different distributions
- Different from subset ranking
  - Sampling of documents for each query is considered.

Elements in two-layer sampling are neither independent nor identically distributed.

Generalization Bound

Theorem 1. Suppose l is the loss function for ranking, $\mathcal{F}$ is the function class of the ranking model, and $l^\circ\mathcal{F}$ is bounded by M, and Rademacher average $E[\mathcal{R}_m(l^\circ\mathcal{F})]$ is bounded by $D(l^\circ\mathcal{F}, m)$, then for arbitrary sample distribution, for $\forall f \in \mathcal{F}$, with probability at least $1-\delta$,

$$R^l(f) \leq \hat{R}^l_{m_1,\ldots m_n}(f) + D(l^\circ\mathcal{F}, n) + \sqrt{\frac{2M^2\log(\frac{4}{\delta})}{n}} + \frac{1}{n}\sum_i D\left(l^\circ\mathcal{F}, \left\lfloor\frac{m_i}{2}\right\rfloor\right) + \sqrt{\sum_i \frac{2M^2\log(\frac{4}{\delta})}{m_i n^2}}.$$

With fixed total budget of labeling ($\sum_i m_i = C$), when the ranking function class satisfies $VC(\mathcal{F}) = V$ and $|f(x)| \leq B$, the optimal tradeoff between number of queries and number of document per query (shallow or deep) is:

$$n^* = \frac{c_1\sqrt{V} + \sqrt{2\log(\frac{4}{\delta})}}{c_1\sqrt{2V}}\sqrt{C}; \quad m_i^* \equiv \frac{C}{n^*}$$

Generalization Bound

Theorem 1. Suppose l is the loss function for ranking, $\mathcal{F}$ is the function class of the ranking model, and $l^\circ\mathcal{F}$ is bounded by M, and Rademacher average $E[\mathcal{R}_m(l^\circ\mathcal{F})]$ is bounded by $D(l^\circ\mathcal{F}, m)$, then for arbitrary sample distribution, for $\forall f \in \mathcal{F}$, with probability at least $1-\delta$,

$$R^l(f) \le \hat{R}^l_{m_1,\dots,m_n}(f) + D(l^\circ\mathcal{F}, n) + \sqrt{\frac{2M^2\log(\frac{4}{\delta})}{n}} + \frac{1}{n}\sum_i D\left(l^\circ\mathcal{F}, \left\lfloor\frac{m_i}{2}\right\rfloor\right) + \sqrt{\sum_i \frac{2M^2\log(\frac{4}{\delta})}{m_i n^2}}.$$

With fixed total budget of labeling ($\sum_i m_i = C$), when the ranking function class satisfies

For more complicated ranking function class, more documents per query and therefore a deep training set is preferred; For simpler ranking function class, a shallow training set is preferred.

How to Get There…

$$Test\ Measure \le Training\ Loss + \varepsilon(n, m, \mathcal{F})$$

Test Measure $\le$ Test Loss $\le$ Training Loss $+$ ε

1

2

Loss Function vs. Ranking Measure

Very different mathematical forms

- Loss Function in ListMLE, as an example

$$L(f;\mathbf{x},g) = -\log P_{PL}(\pi_g | f(\mathbf{x}))$$

Based on the scores $f(\mathbf{x})$ produced by the ranking model.

- 1- NDCG (Normalized Discounted Cumulative Gain)

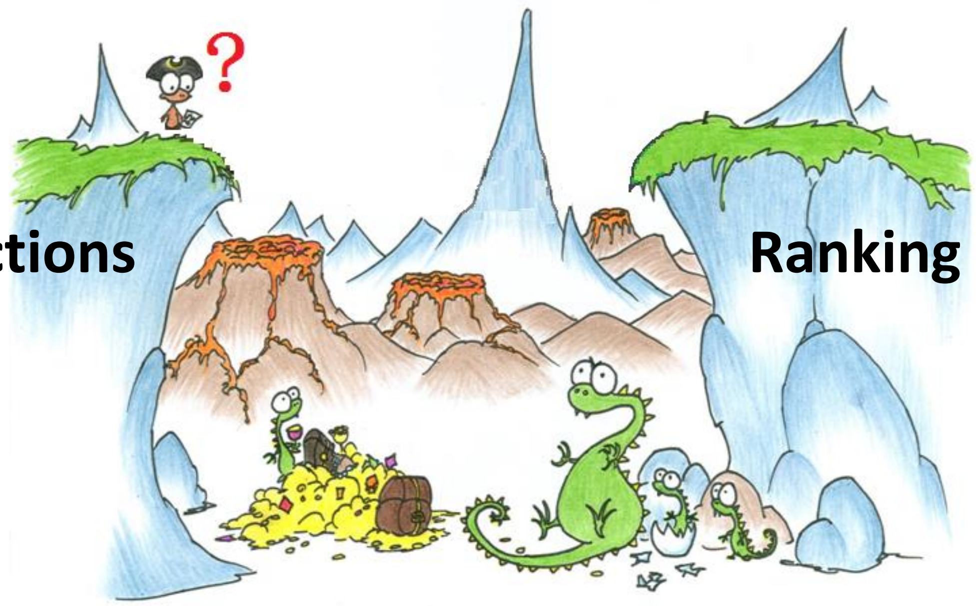
$$NDCG@K = Z_k \sum_{k} G(\pi^{-1}(k))/\log(k+1)$$

Based on the ranked list π by sorting the scores.

Challenge

- Relationship between loss and measure in ranking is unclear due to their different mathematical forms.

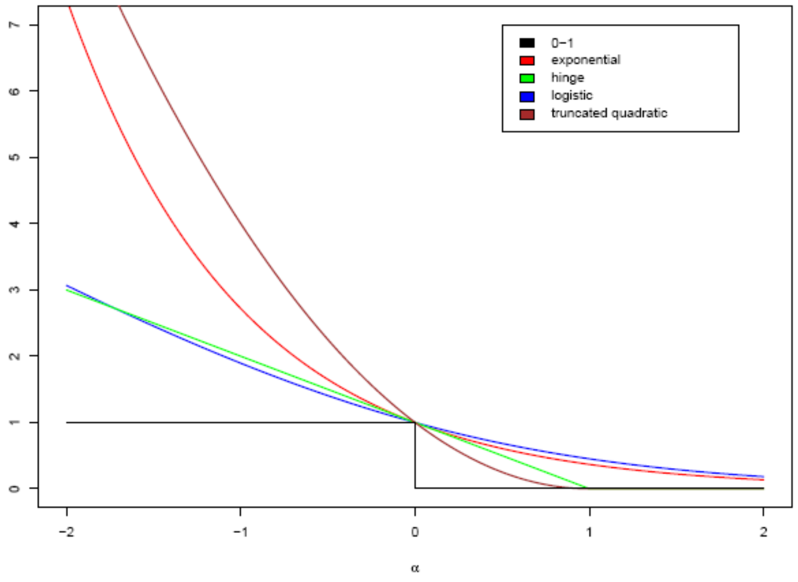
- .



Challenge

- Relationship between loss and measure in ranking is unclear due to their different mathematical forms.

- In contrast, for classification, both loss and measure are defined regarding individual documents and their relationship is clear.



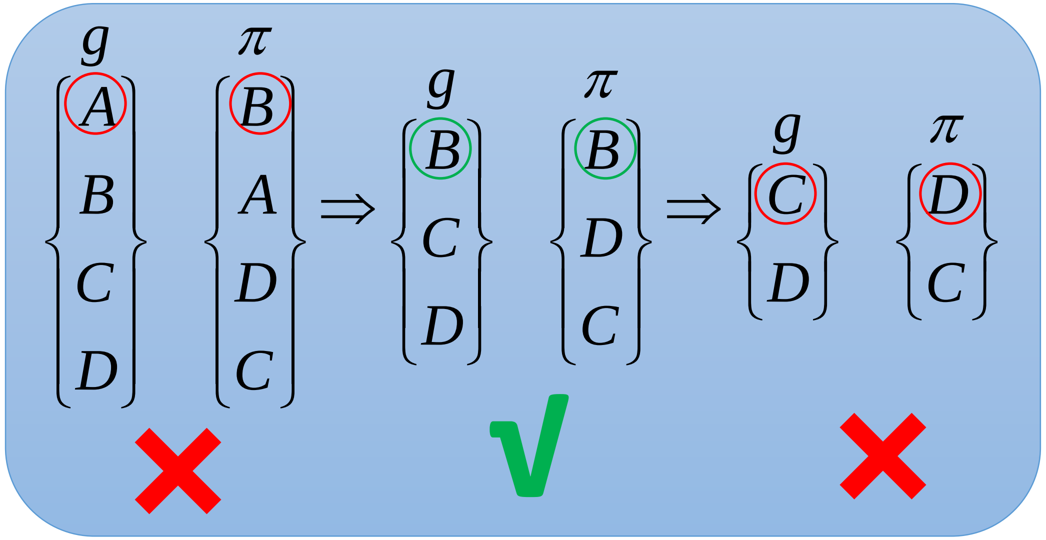
Essential Loss for Ranking

[Chen et al. NIPS 2009]

- Model ranking as a sequence of classifications

Ground truth permutation: $g = \{A \succ B \succ C \succ D\}$

Prediction of the ranking function f: $\pi = \{B \succ A \succ D \succ C\}$



Classifier
$T_f(\mathbf{x}_{(s)})$

Output the document with the largest ranking score

$$L_\beta(f;\mathbf{x},g) = \sum_s \beta(s) I_{\{T_f(\mathbf{x}_{(s)}) \neq g(s)\}}$$

The weighted classification error for each step in the sequence

Essential Loss vs. Ranking Measures

Theorem 2. Given K-level rating data with n_k objects with rating k, the following inequalities hold for $\forall f$:

$$1)\ 1 - NDCG(f;\mathbf{x},g) \le Z_n L_{\beta_1}(f;\mathbf{x},g), \text{ where } \beta_1(s) = \frac{G\left(g(\pi^{-1}(s))\right)}{\log(s+1)}.$$

$$2)\ 1 - AP(f;\mathbf{x},g) \le \frac{1}{\sum_{i \ge k^*} n_i} L_{\beta_2}(f;\mathbf{x},g), \text{ where } \beta_2(s) \equiv 1.$$

Essential Loss vs. Loss Functions

Theorem 3. The loss functions of many learning to rank methods including RankSVM, RankBoost, and ListMLE, are all upper bounds of the essential loss, i.e., for $\forall f$:

$$L_\beta(f;\mathbf{x},g) \leq \frac{\max_s \beta(s)}{ln2} L(f;\mathbf{x},g)$$

Overall Theory

- (1-NDCG), (1-MAP) ≤ Essential Loss ≤ Loss Functions

$$1 - NDCG(f;\mathbf{x},g) \le Z_n L_{\beta_1}(f;\mathbf{x},g) \le \frac{\max_s \beta_1(s) Z_n}{ln2} L(f;\mathbf{x},g)$$

$$1 - AP(f;\mathbf{x},g) \le \frac{1}{\sum_{i \ge k^*} n_i} L_{\beta_2}(f;\mathbf{x},g) \le \frac{1}{ln2 \sum_{i \ge k^*} n_i} L(f;\mathbf{x},g)$$

Summary of Part I

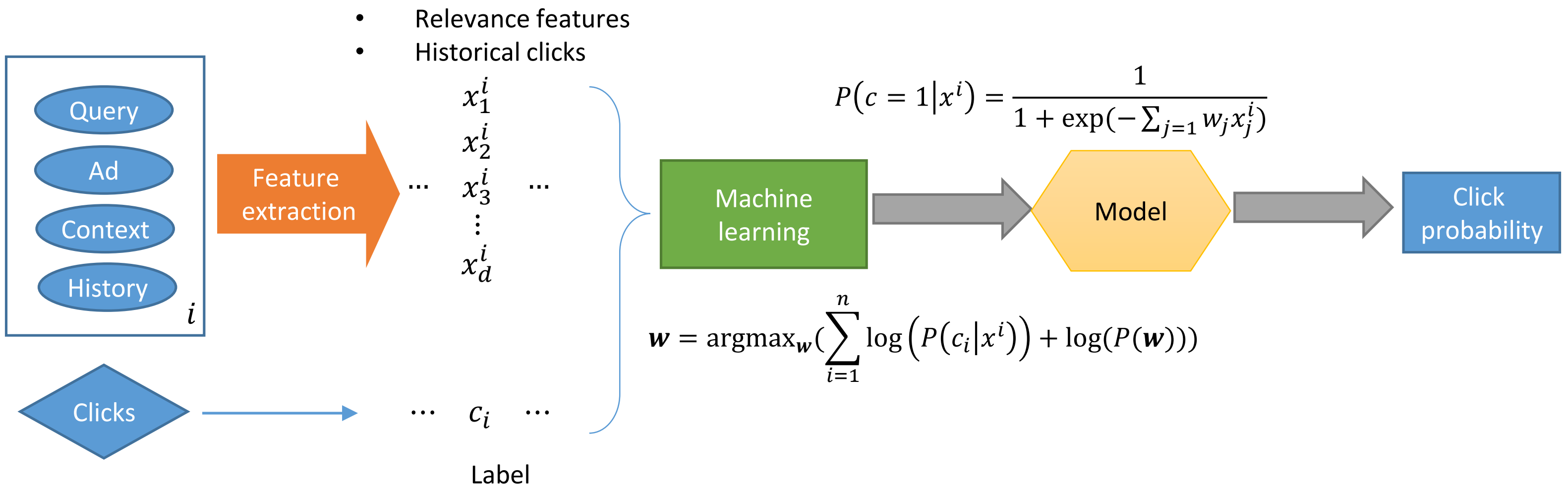
- Learning to rank methods
  - Better to treat ranking as a new machine learning problem, rather than simply reducing it to regression, classification, or pairwise regression
  - Both listwise loss function and ranking functions could be employed

- Generalization theory for ranking
  - Essential loss plays as a bridge between ranking measures and widely used loss functions for ranking
  - Minimization of widely used loss functions can effectively maximize ranking measures

Part II: Machine Learning for Ad Auctions

Revisit of Auction Mechanism

- GSP auction
  - Ranks ads by $pClick \times bid$
  - Charges clicked ads by $\frac{pClick_{next} \times bid_{next}}{pClick}$.
- Naive application of machine learning: learn $pClick$ then follow GSP



Microsoft

What’s the Problem?

- User click behavior is more complicated than pointwise classification
- Advertiser behavior is totally missing in the approach
- GSP auction is not a data driven model

User Click Behavior Prediction

- Psychological click prediction (KDD 2013)
- Relational click prediction (WSDM 2012)
- Temporal click prediction (AAAI 2014)

Advertiser Bidding Behavior Modeling

- Fictitious play model (WINE 2012)
- Rationality model (WWW 2013)
- Markov Behavior Model (AAAI 2014)

Auction Mechanism Design

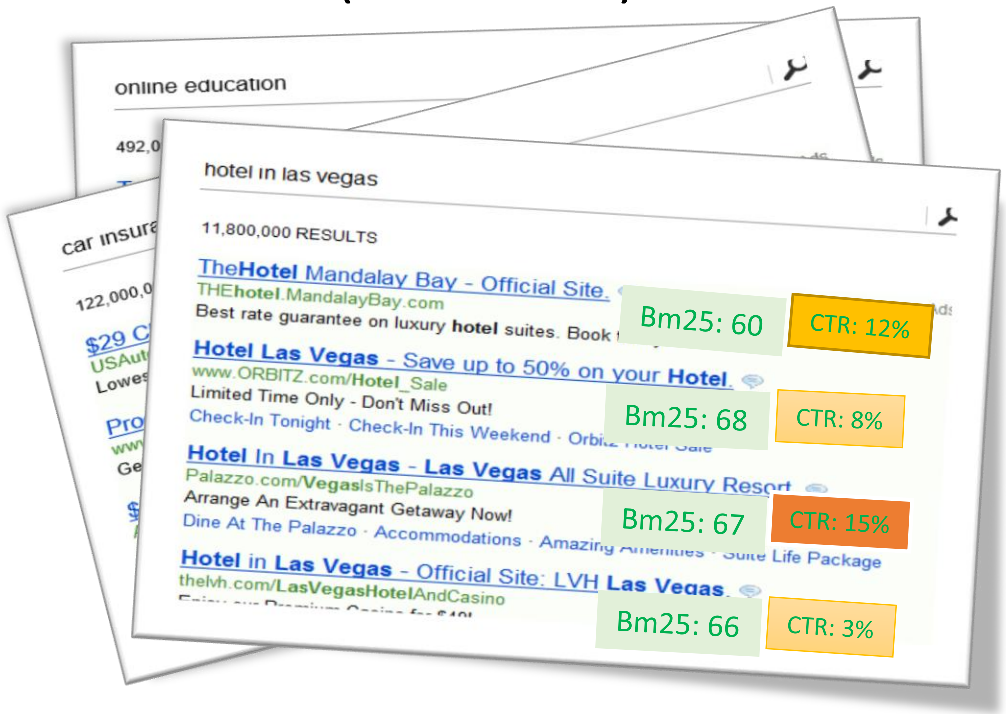
- Game-theoretic machine learning for mechanism optimization (IJCAI 2013)
- Auction with value externality (AAMAS 2014)
- Broad-match GSP auction (EC 2014)

1. User Click Behavior Modeling

- Psychological click prediction
  - Beyond modeling “what (relevance)” and “how (historical clicks)”, model “why” users click on ads
- Relational click prediction
  - Instead of considering ad impressions as i.i.d. samples, model externality between different ads
- Temporal click prediction
  - Instead of considering ad impressions as i.i.d. samples, model temporal dependency between different clicks of the same user on the same ad

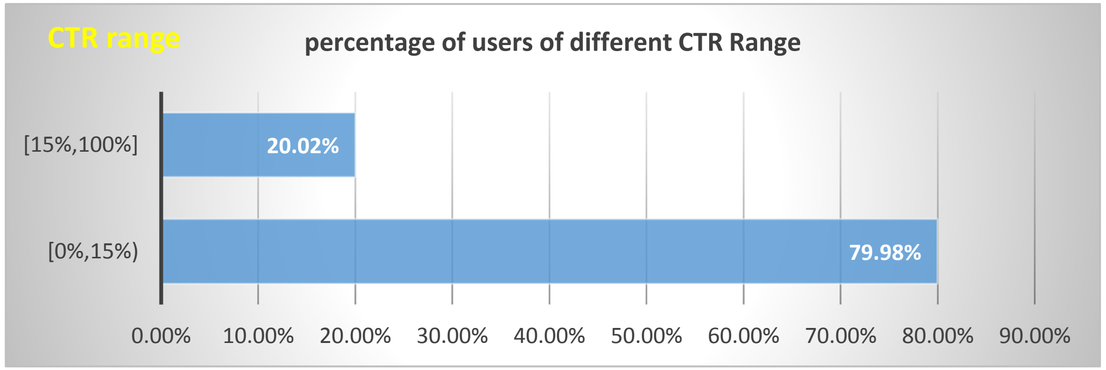
Why Users Click?

What (Relevance)



Average Pearson correlation between relevance score (BM5) and click is only 0.08.
(Very weak, almost equivalent to random guess!)

How (Historical Clicks)



Given the same ad, the variance of CTRs across different users is very large. Difficult to predict one user's click behavior based on other users.

(Dataset: production pClick training data set with 15M ad impression, ~2M query event)

Microsoft

Why Users Click?

Query: **Nike**

Free Nike Coupons
Download And Print Nike Coupons (100% Free)
0.08

Nike - Sale Prices
Latest Fashions and Styles on Sale. Buy Nike Fast!
0.005

AKADEMA Baseball Outlet
PRO, ROOKIE, FASTPITCH, APPAREL BATS, MITT & GLOVES $7.99 - 199.99
0.003

Discount, Free

Query: **Perfume**

Perfume.com official site
10,000+ brand name perfumes and colognes - up to 80% off retail!
0.166

Luxury English Perfume
Shop online for luxury perfumes for men, women & the home.!
0.005

Versace Perfume
The Scent of You. Discover Versace Perfume!
0

Trust, Brand

Query: **HP Drivers Download**

HP Drivers Download
(Recommended) Download HP Drivers. Download HP Drivers in Seconds.
0.484

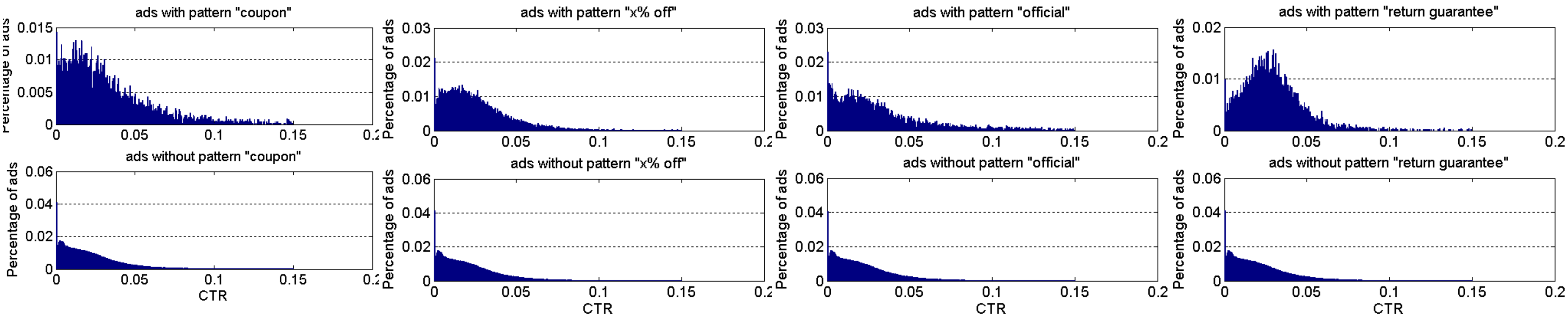
HP Drivers Download
Free Download: HP Drivers Update. Download & Install HP Drivers Now
0.167

HP Drivers Downloads
(Recommended) HP Printer Drivers. HP Drivers Download Center.
0.103

Fast, Convenient

Why Users Click?

Patter	Coverage	CTR Lift
“coupon”	2.2%	+47.5%
“x% off”	4.1%	+19.7%
“official”	2.6%	+25.0%
“return guarantee”	1.9%	+31.4%

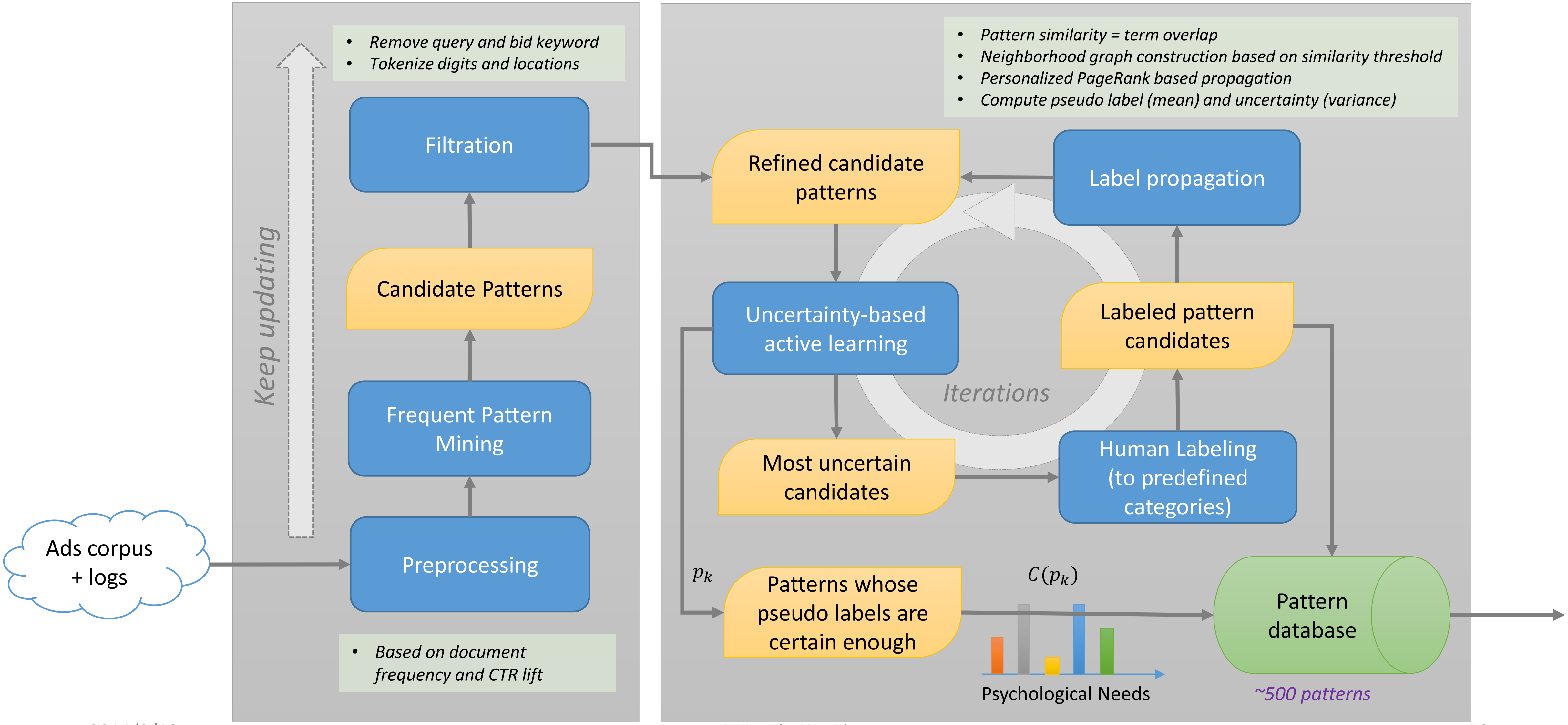


Generalization: Users' Psychological Needs

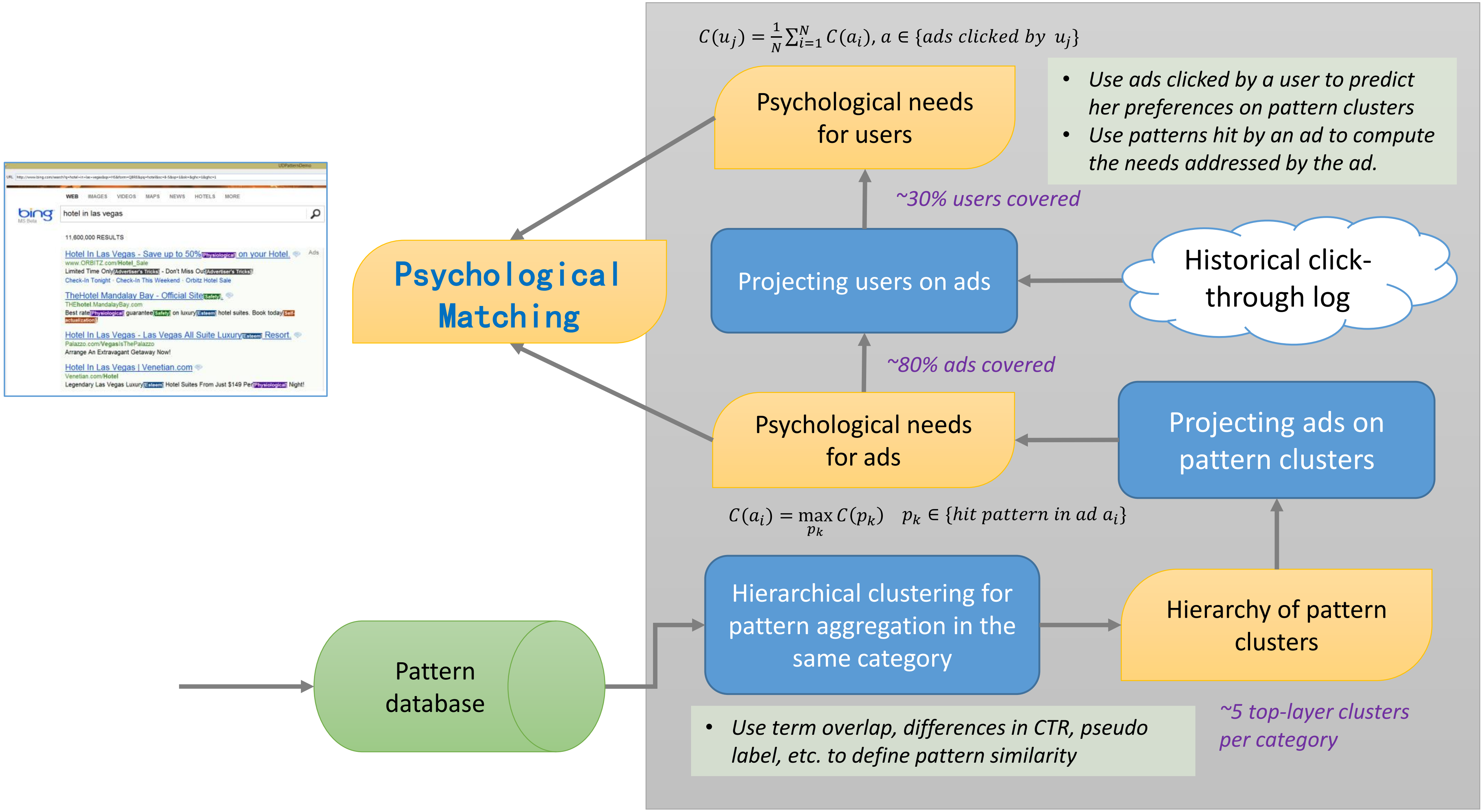
Self-actualization	Advance your career	Your dream	Achieve yours	Your ideal	Moments of yours		
Esteem	First Class	Top Quality	VIP	Ultimate Experience	Just for you	Unique Specialty	Top Brands
Love/belonging	Find it Nearby	Call 888-888-8888	Visa, Amex, Paypal Accepted	Hotels in Chicago	Same Day Shipping		
Safety	365 Days Return	100% Guaranteed	Official site	20,000+ PCs & Laptops	Customer Reviews		
Physiological	Save money	Best price	30% off	Coupon	Free Shipping	20% Discount	

Maslow's hierarchy of needs

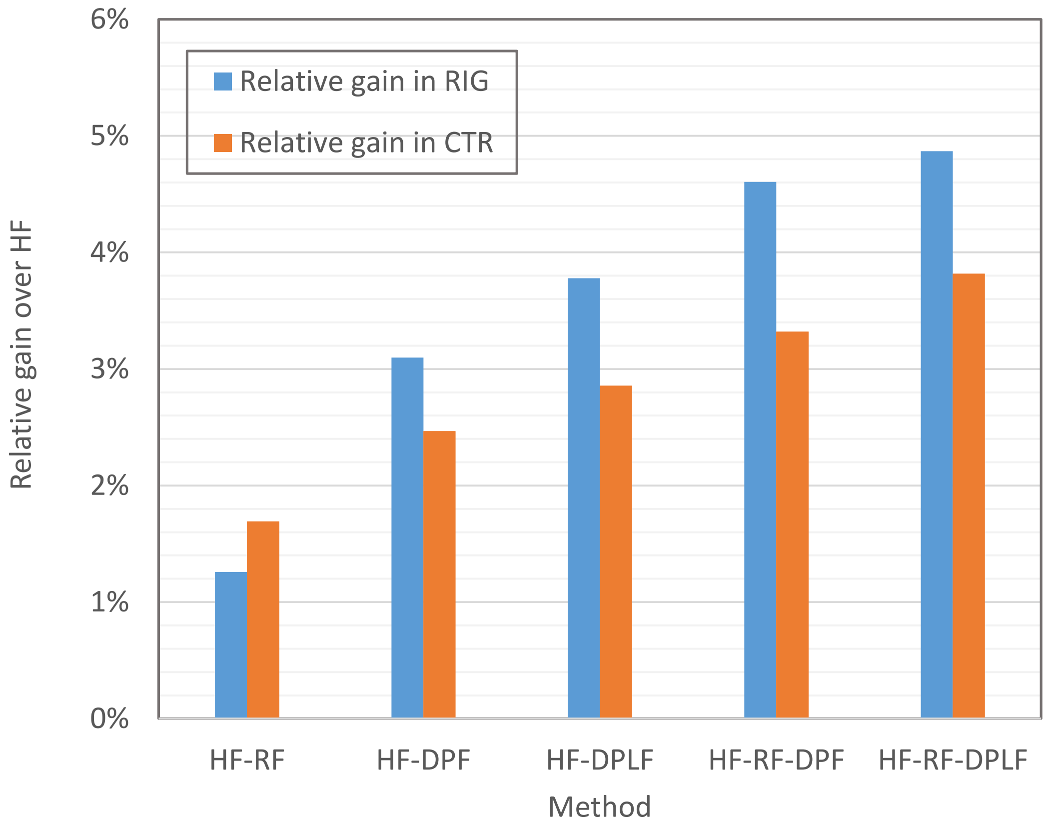
Psychological Needs Mining [Wang et al. KDD 2013]



Psychological Needs Mining [Wang et al. KDD 2013]



Click Prediction based on Psychological Features



- **HF**: only uses historical click features.
- **HF-RF**: uses historical click features and relevance features.
- **HF-DPF**: uses historical click features and desire pattern features.
- **HF-DPLF**: uses historical click features and both desire pattern and desire level features.
- **HF-RF-DPF**: uses historical click features, relevance features, and desire pattern features.
- **HF-RF-DPLF**: uses historical click features, relevance features, and both desire pattern and desire level features.

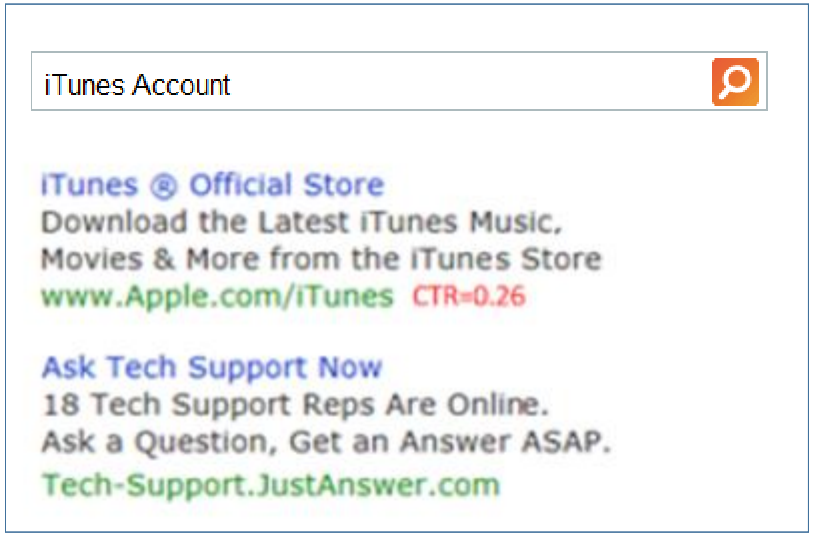
4-5% CTR increase corresponds to hundreds of millions of revenue increase…

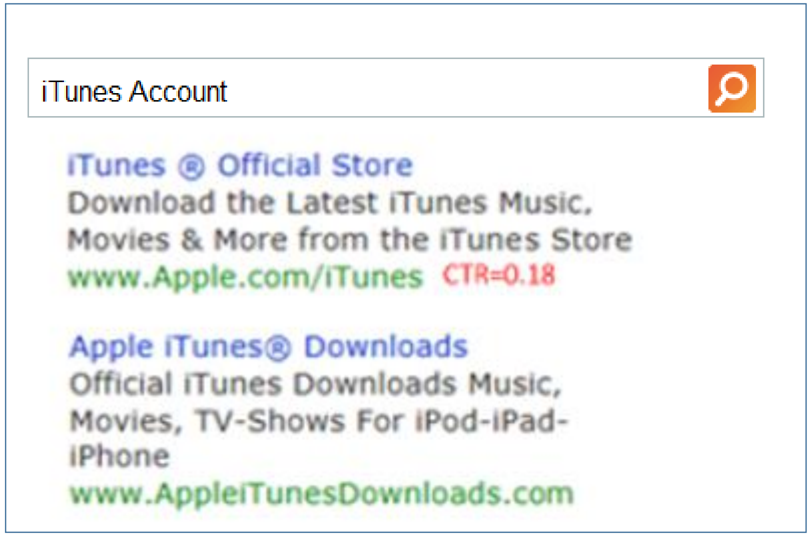
1. User Click Behavior Modeling

- Psychological click prediction
  - Beyond modeling “what (relevance)” and “how (historical clicks)”, model “why” users click on ads
- Relational click prediction
  - Instead of considering ad impressions as i.i.d. samples, model externality between different ads
- Temporal click prediction
  - Instead of considering ad impressions as i.i.d. samples, model temporal dependency between different clicks of the same user on the same ad

Externality between Ads

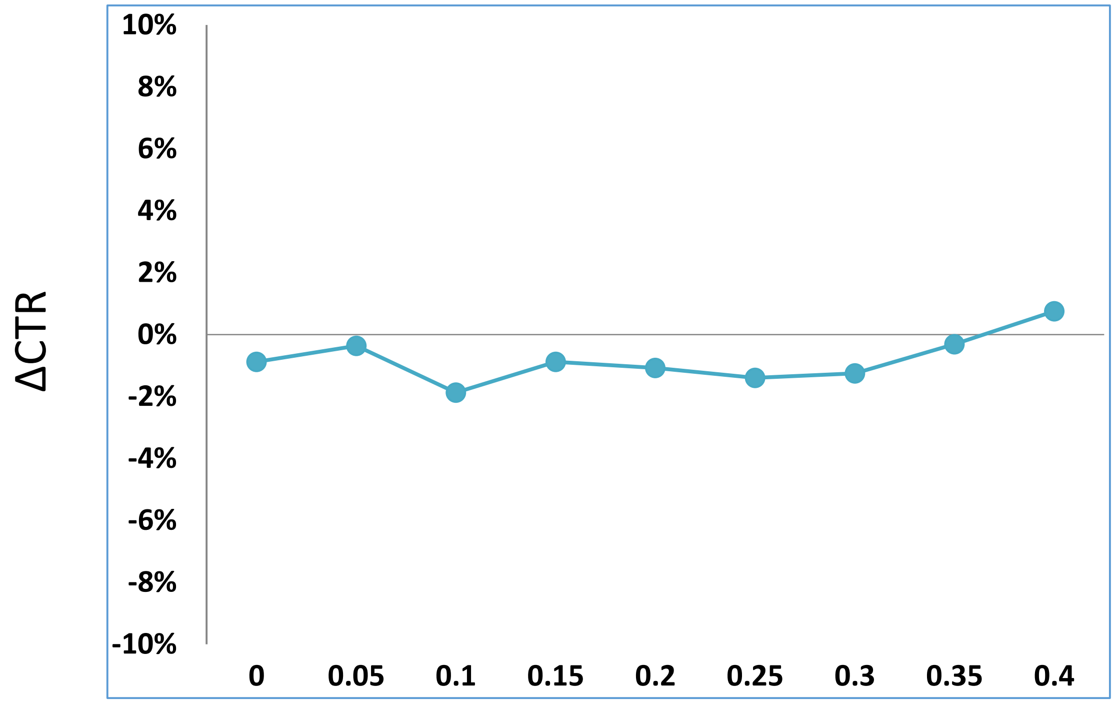
- An ad has lower CTR if shown together with high-quality (CTR) ads





Quality as Externality

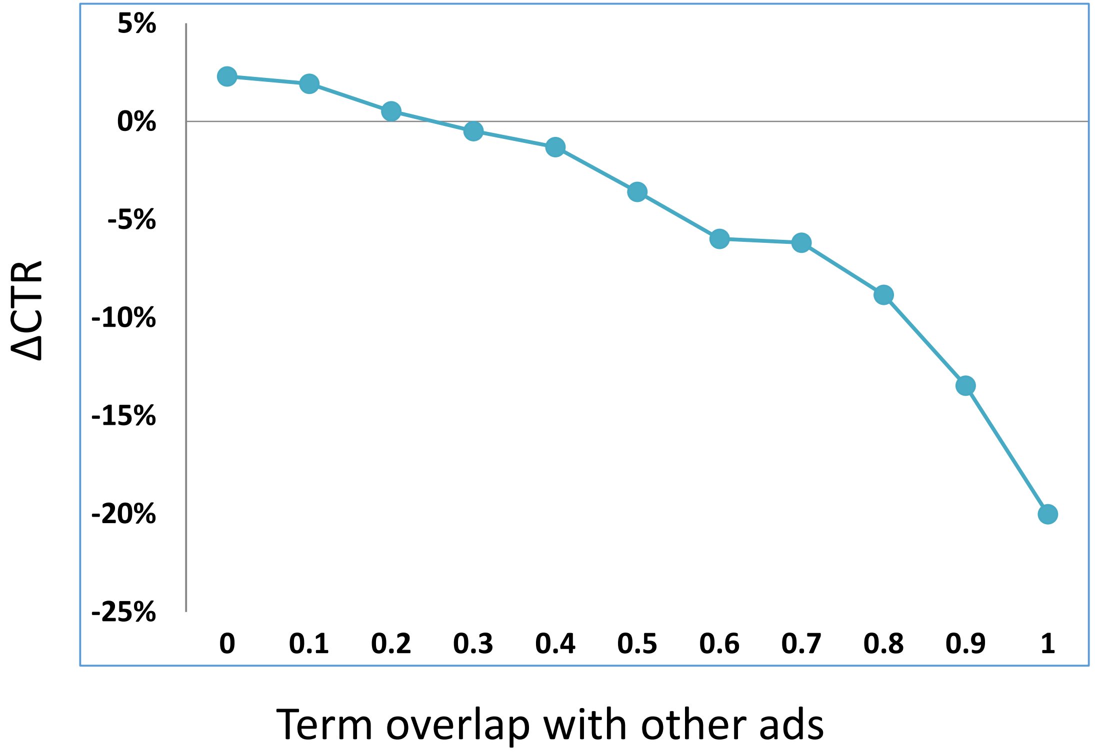
- Not as obvious as expected



Average CTR of surrounding ads

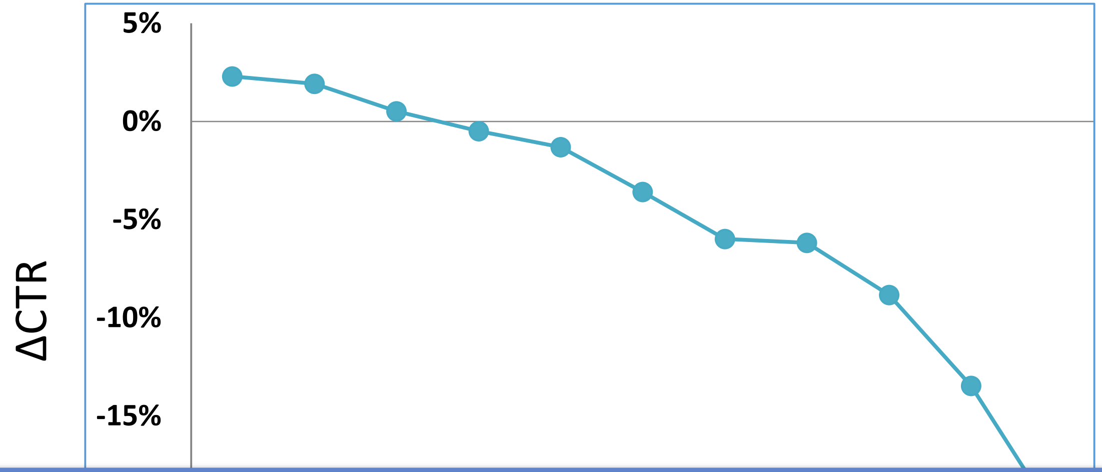
Similarity as Externality

- Clearer trend observed



Similarity as Externality

- Clearer trend observed



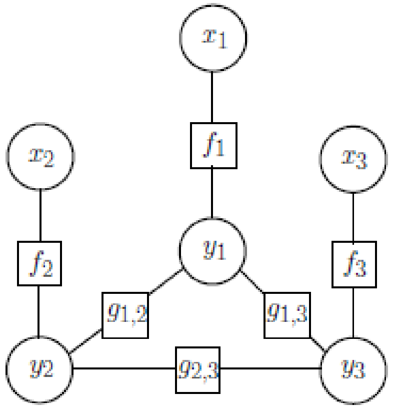
Similarity is strongly negatively correlated with ΔCTR: Pearson correlation coefficient is -0.943.

Continuous Conditional Random Fields

[Xiong, et al. WSDM 2012]

- Conditional probability:
  - Given X, input feature vectors of ads and Y, the log mod of CTR

$$P(Y|X) = \frac{1}{Z(X)} \exp\{\sum_i \underline{h(y_i, X; \omega)} + \sum_{j>i} \underline{\beta g(y_i, y_j, X)}\}$$



Vertex feature function

$$h(y_i, X; \omega) = -((y_i - \mathrm{f}(x_i; \omega))^2$$

Model the single ads feature x_i

Edge feature function

$$g(y_i, y_j, X) = -s_{i,j}(y_i + y_j)$$

$S_{i,j}$ means the similarity between ads i and ad j.

Learning and Inference

- Maximum likelihood estimation for learning

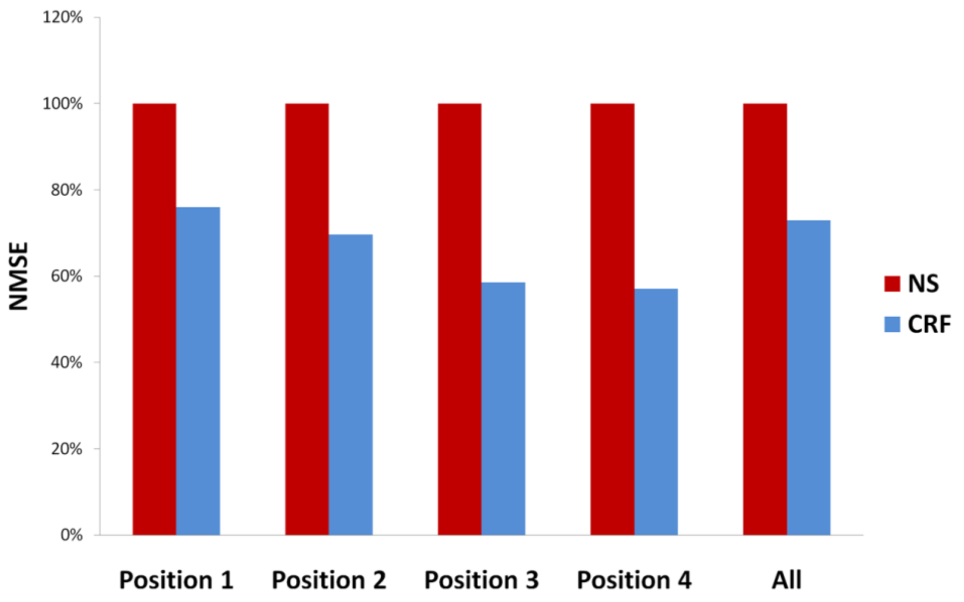
$$L(\theta) = \sum_{q=1}^{N}\left(-\sum_{i}(y_i^q - f(x_i^q; w))^2 - \beta \sum_{j>i} s_{i,j}(y_i^q + y_j^q) - logZ(X^q)\right)$$

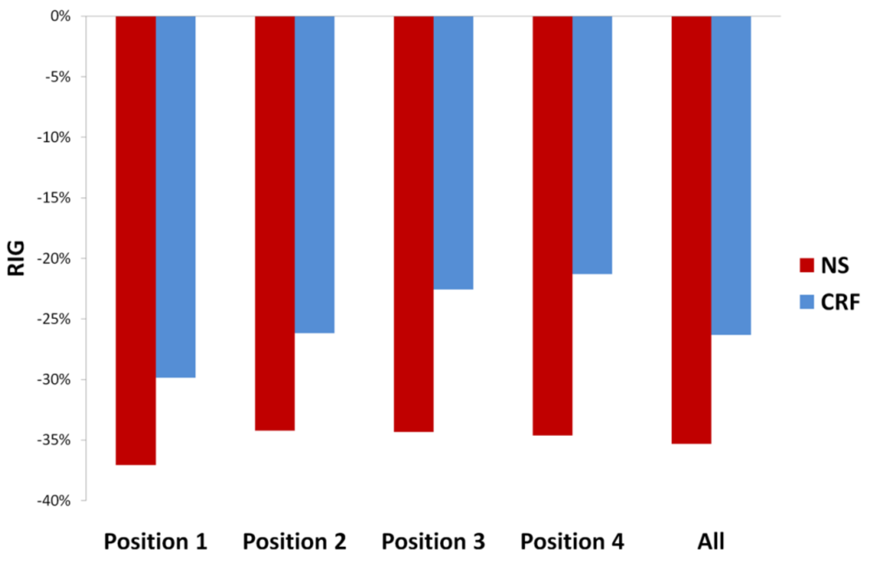
- During inference, find the Y that maximizes the conditional probability:

$$Y^* = argmax_Y P(Y|X;\theta)$$
$$= argmax_Y \left\{-\big(Y - f(X;w)\big)^T \big(Y - f(X;w)\big) - \beta S^T Y\right\}$$

$P(Y|X;\theta)$ is concave w.r.t. Y, and maximization can be efficiently performed.

Experimental Results





NS: Logistic regression, no similarity information considered

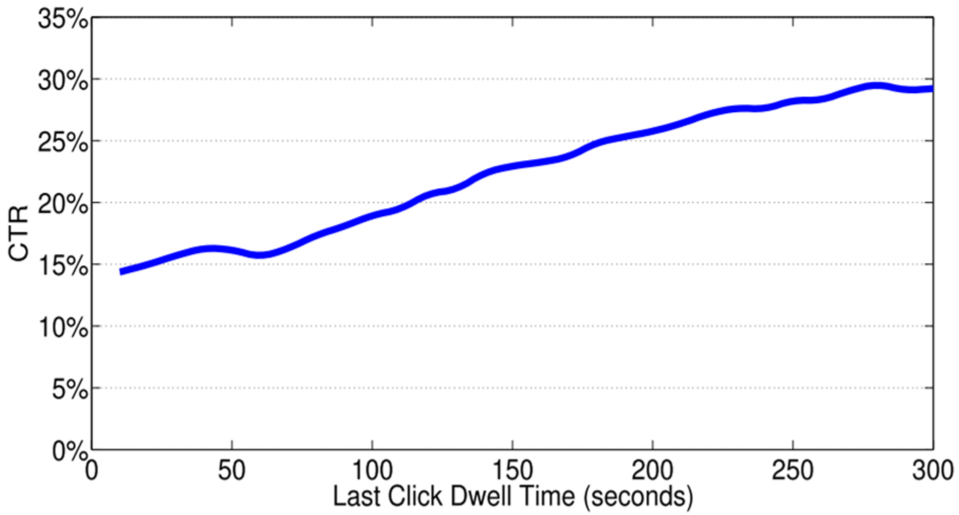
- Reduces MSE by more than 50% as compared to baseline
- Increases RIG by more than 40% as compared to baseline

1. User Click Behavior Modeling

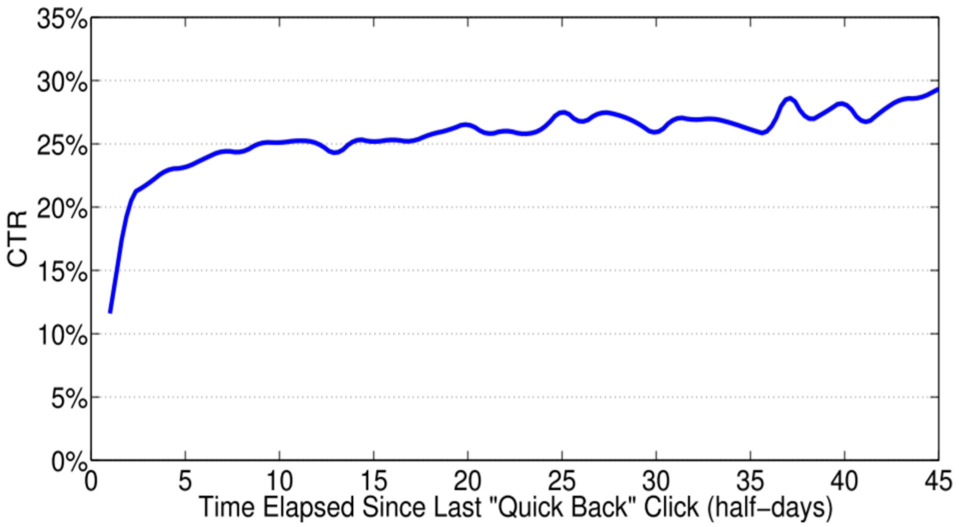
- Psychological click prediction
  - Beyond modeling “what (relevance)” and “how (historical clicks)”, model “why” users click on ads
- Relational click prediction
  - Instead of considering ad impressions as i.i.d. samples, model externality between different ads
- Temporal click prediction
  - Instead of considering ad impressions as i.i.d. samples, model temporal dependency between different clicks of the same user on the same ad

Temporal Dependency Matters

- Last click dwell time influences next click-through rate
  - Longer dwell time on the ad landing page tends to results in higher future click-through rates.



- Vanishing impact of “quick back” click (short dwell time) on future clicks
  – Users have limited memory, and tend to gradually forget unhappy experience along with the time.



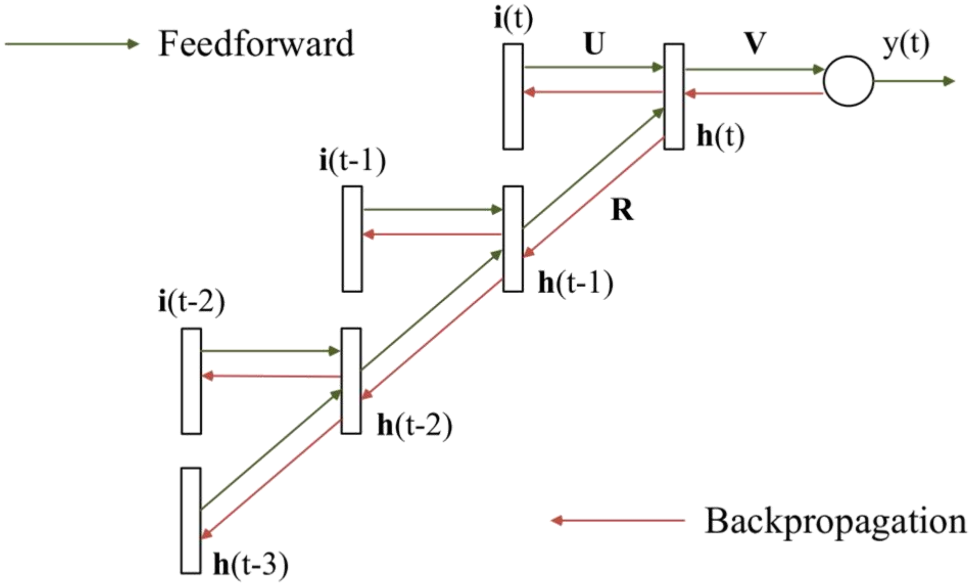
Classical Model of Temporal Dependency

- Time series analysis
  - Focus on modeling trends or periodicity in time sequences
  - However, temporal dependency between click behaviors is complex and dynamic: varying timespan, multi-type, high-order, personalized.
- New model is needed – Recurrent Neural Networks (RNN)
  - RNN is a proven tool to model complex dependency in sequential data in different applications, e.g., RNN language model, RNN based handwriting/speech recognition.

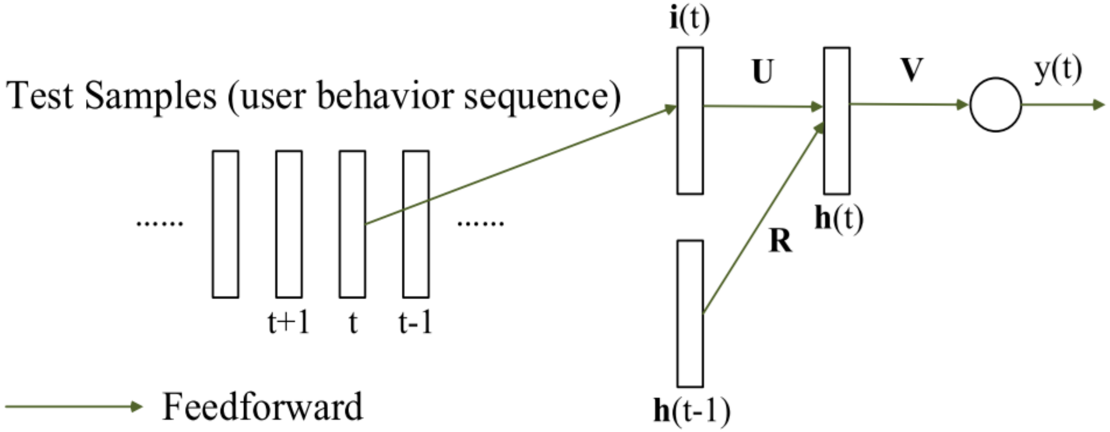
RNN Model for Click Prediction

[Zhang et al. AAAI 2014]

- Training



- Test

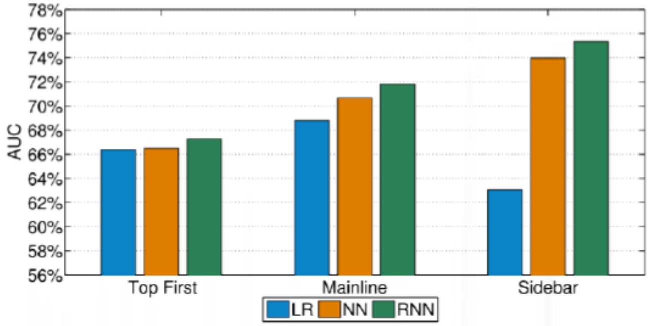


Experimental Results

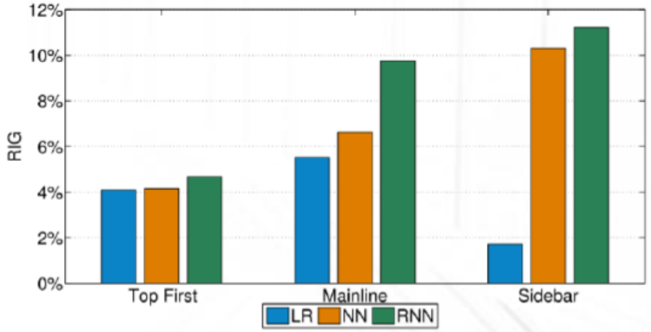
- Settings
  - Click-through logs from Bing.com
  - One week for training and the second week for test

- Experimental results
  - In terms of AUC:
    - About 1.7% relative gain over logistic regression
    - About 0.5% relative gain over neural networks
  - In terms of RIG:
    - About 17.3% relative gain over logistic regression
    - About 10.0% relative gain over neural networks

Model	AUC	RIG
LR	87.48%	22.30%
NN	88.51%	23.76%
RNN	**88.94%**	**26.16%**



(a) AUC on specific ad positions



(b) RIG on specific ad positions

2. Advertiser Bidding Behavior Modeling

Machine Learning

i.i.d. behaviors

- Ignore strategic behavior of advertisers; assume previous and future behaviors follow same distribution
- The distribution is independent of auction mechanism.

Reality

Self interest + bounded rationality

- After each auction, advertisers get access to partial information I_t (#clicks, cost per click)
- Given information I_t and current bids b_t, advertisers change their bids to b_{t+1}.

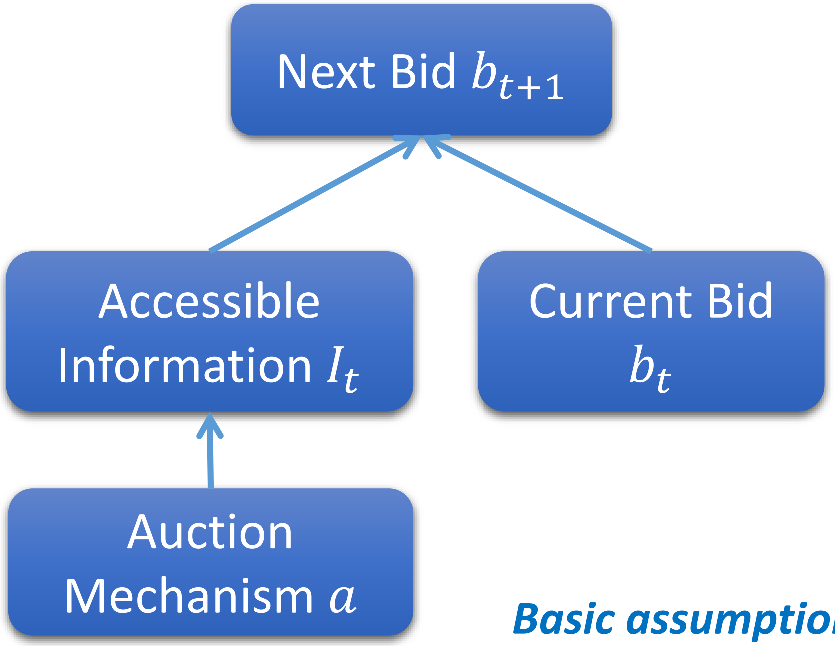
Game Theory

Self interest + full rationality

- Well-defined utility + full information + capability of utility maximization
- Best response model, quantal response model (probability proportional to utilities), etc.

How to Learn Advertiser Behaviors?

- More appropriate assumption: Markov Behavior Model
  [Tian et al. AAAI 2014]



- Given accessible information I_t and current bids b_t, bidding behaviors of advertisers can be modeled by a Markov transition matrix indicating how likely they change bid from b_t to b_{t+1}:
- $P(b_{t+1}|all\ info) = P\left(b_{t+1}\middle|I_t\ , b_t\ \right) = \prod_{i=1}^{n} P_i(b_{t+1}^i|I_t^i, b_t^i)$.

***Basic assumption**: advertisers only have limited memory – their future bidding behaviors only depend on the previous information in a finite time period.*

Generality

- The Markov model can cover most previous behavior models used in game theory and machine learning literature

Best response model	$P(b_{t+1}^i \mid I_t, b_t^i) = \mathbb{I}_{\{b_{t+1}^i = \arg\max_b U^i(I_t, b)\}}$
Quantal response model	$P(b_{t+1}^i \mid I_t, b_t^i) \propto U^i(I_t, b_{t+1}^i)$
Random model	$P(b_{t+1}^i \mid I_t, b_t^i) = \frac{1}{n}$
I.I.D. model	$P(b_{t+1}^i \mid I_t, b_t^i) = \mathbb{I}_{\{b_{t+1}^i = b_t^i\}}$

Aymptotical Stability

- It can be proven that the system with a general Markov behavior model will be stable with a stationary distribution.
- The Markov behavior model can cover specific dynamic behaviors studied in the literature of game theory, which lead to certain equilibrium.

Best response model

$$P(b_{t+1}^i | I_t, b_t^i) = \mathbb{I}_{\{b_{t+1}^i = \arg\max_b U^i(I_t, b)\}}$$

→ Nash Equilibrium

Quantal response model

$$P(b_{t+1}^i | I_t, b_t^i) \propto U^i\left(I_t, b_{t+1}^i\right)$$

→ Quantal response Equilibrium

Learnability

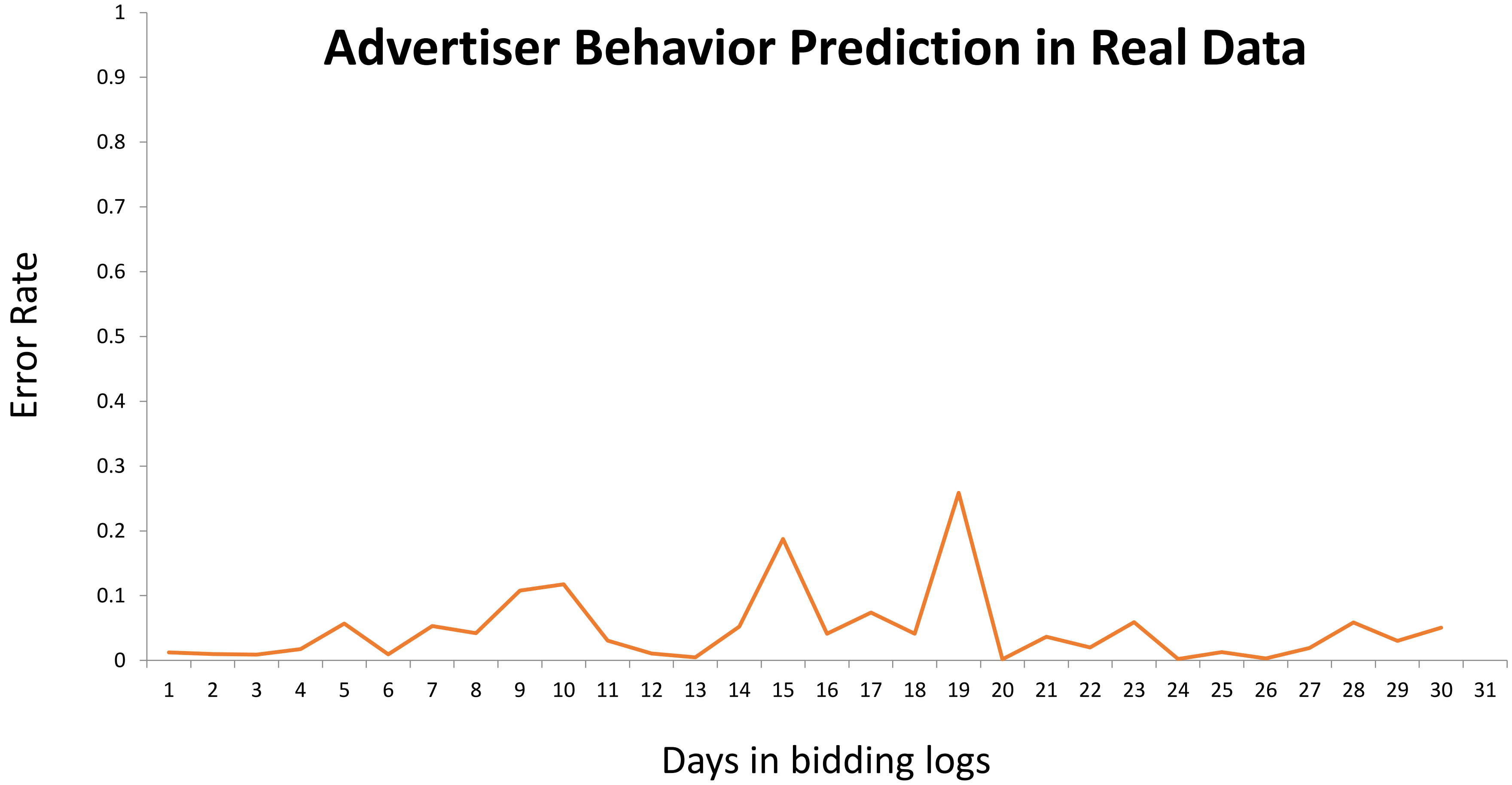
Parametric Learning

- Assume the transition probability to take a certain parametric form, e.g.,
  - $P_i(b'|I,b) = P_i(b|f(\omega; I, b))$,
- Given training data with size T_1, learn the parameters by means of maximum likelihood estimation:
  - $B_{T_1} = \arg\min_{B \in \mathcal{B}} l(B; a_0, u_1, \ldots, u_{T_1})$

Non-Parametric Learning

- Directly estimate the transition probability $P_i(b'|I,b)$
- Given training data with size T_1, estimate $P_i(b'|I,b)$ by the conditional frequency in the training data:
  - $\widehat{P}_i(b'|I,b) = \frac{\sum_{t=1}^{T_1} \mathbb{I}_{\{b_{t+1}=b', b_t=b, h_t=H_j\}}}{\sum_{t=1}^{T_1} \mathbb{I}_{\{b_t=b, h_t=H_j\}}}$

Accuracy



Advertiser Behavior Prediction in Real Data
Error Rate
Days in bidding logs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

3. Auction Mechanism Optimization

- Conventional machine learning approach

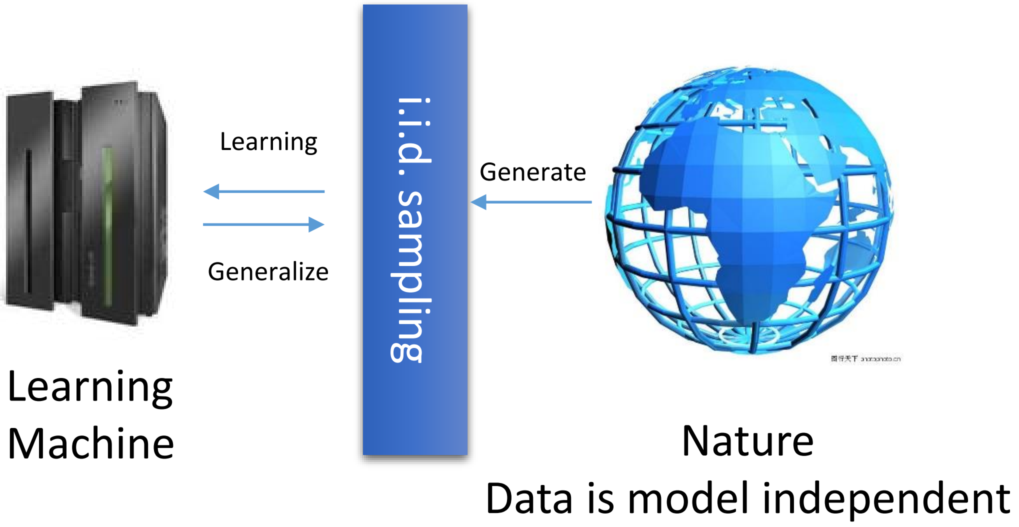
Define Objective Function → Optimize objective function on historical bidding data based on i.i.d. assumption → Use the learned mechanism to rank and price ads in test data

- Example: RankLogistic [Zhu et al. SIGIR 2009]
  - Training data: historical auction logs (queries, clicks, bids)
  - Objective function: empirical revenue (e.g., for first price auction with single slot)

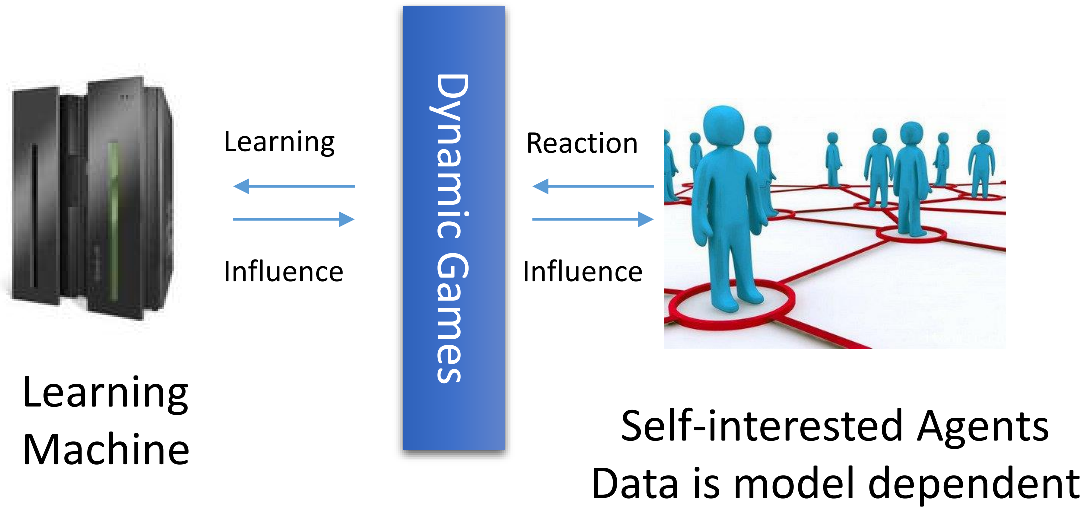
$$R(w) = \sum_{q \in Q} \sum_{p=1}^{n_q} rev_q(p) c_q(p) \mathbb{I}_{\left\{\min_{i \neq p}\left\{f\left(w, x_q(p)\right) - f\left(w, x_q(i)\right)\right\} > 0\right\}}$$

What’s the Problem?

Ideal assumption: model will not affect data distribution; i.i.d. sampling guarantees generalization.

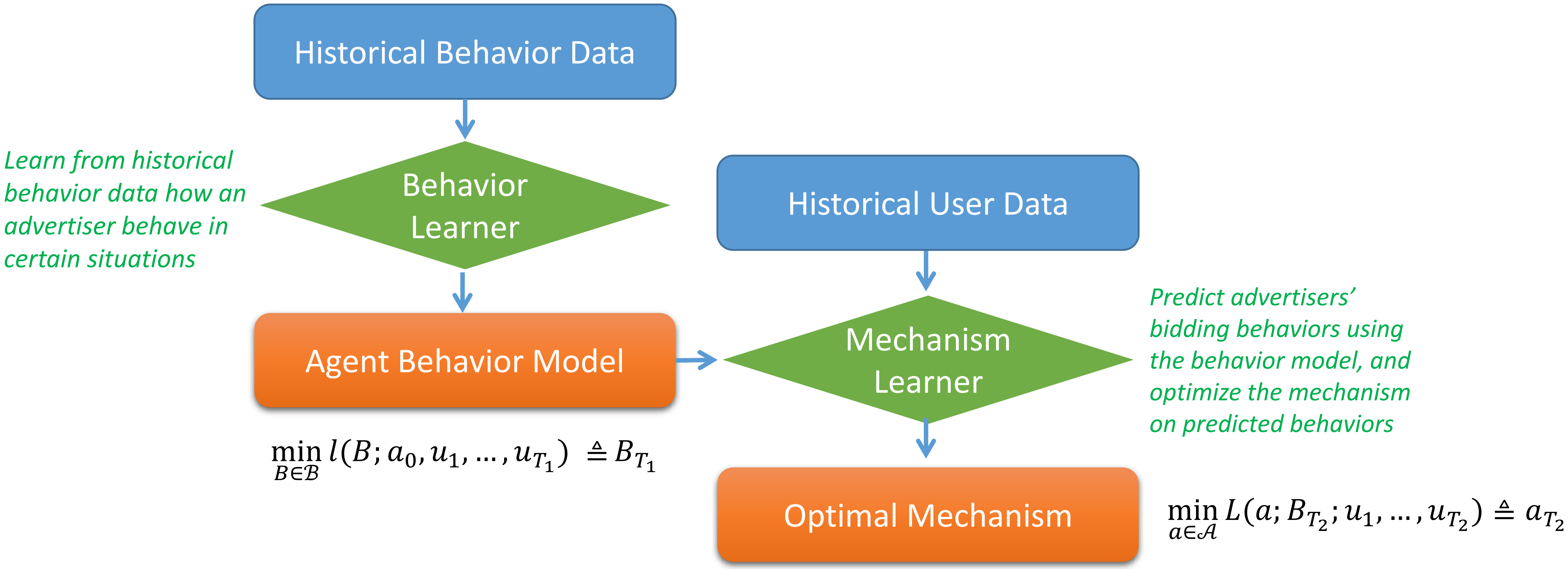


Real situation: agents are strategically behaving in response to model, resulting in non-i.i.d. distribution.



Remove Wrong Assumptions

- Both rationality assumption and i.i.d. assumption are not realistic!
- A new model under Markov assumption is desirable [He et al. IJCAI 2013]



Game-Theoretic Learning

1: Auction Implement

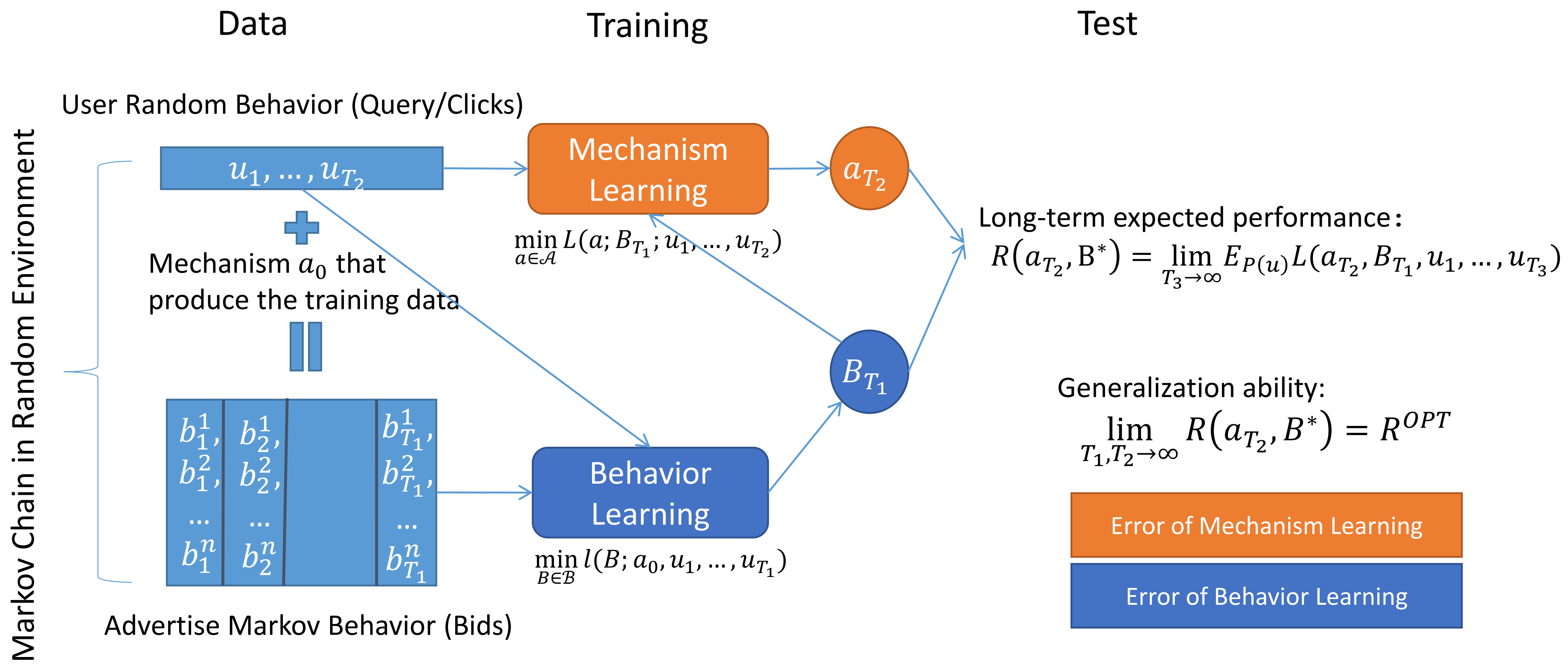
Given query stream u_t and bid profile b_t at time t, and the auction a, run the auction process, compute I_t, and compute the expected search engine revenue: $Rev(a; b_t, u_t)$.

2: Bid Update

Based on I_t, predict bid profile b_{t+1} at time $t+1$ by Markov behavior model B_{T_1}, and make it as input for auction at time $t+1$.

- Implement above process on user data with size T_2, and compute the loss function (i.e., minus average search engine revenue)
  - $L(a; B_{T_1}; u_1, \dots, u_{T_2}) = -\frac{1}{T_2}\sum_{t=1}^{T_2} Rev(a; b_t(B_{T_1}), u_t)$.
- Learn optimal auction mechanism by minimizing the loss function
  - $a_{T_2} = \arg\min_{a\in\mathcal{A}} L(a; B_{T_1}; u_1, \dots, u_{T_2})$, where $\mathcal{A}$ is the auction mechanism space

Generalization Analysis



Generalization Analysis

$$\min_{B\in\mathcal{B}} l(B; a_0, u_1, \dots, u_{T_1}) \triangleq B_{T_1}$$

$$\Downarrow$$

$$\min_{a\in\mathcal{A}} L(a; B_{T_2}; u_1, \dots, u_{T_2}) \triangleq a_{T_2}$$

$$\Downarrow$$

$$R(a_{T_2}, B^*) = \lim_{T_3\to\infty} E_{P(u)} L(a_{T_2}, B^*, u_1, \dots, u_{T_3})$$

Objective: $\lim_{T_1,T_2\to\infty} R(a_{T_2}, B^*) \xrightarrow{p} R^{OPT}$

Error Decomposition

$$R(a_{T_2}, B^*) - R(a^*, B^*) = 2K\sup_{a\in\mathcal{A}} C(a, B^*) \boxed{\left\|B_{T_1} - B^*\right\|_\infty} + 2\sup_{a\in\mathcal{A}} \boxed{\left|R(a_{T_2}, B_{T_1}) - R(a, B_{T_1})\right|}$$

Parametric Method: $\mathrm{P}\left\{\left\|B_{T_1} - B^*\right\|_\infty \geq \epsilon\right\} \leq 2e^{-\frac{\left(T_1\epsilon|\mathcal{B}|^2|\mathcal{H}|\delta_0 - 2C_1N_0\right)^2}{2T_1N_0^2C_1^2}}$

Non-Parametric Method: $\mathrm{P}\left\{\left\|B_{T_1} - B^*\right\|_\infty \geq \epsilon\right\} \leq 2Ce^{-\frac{\left(C_2T_1\delta_0|\mathcal{B}||\mathcal{H}|\epsilon - 2N_0(|\mathcal{B}|+1)\right)^2}{2T_1N_0^2(|\mathcal{B}|+1)^2}}$

Error for Behavior Learning

$$P\left\{\sup_{a\in\mathcal{A}}\left|\mathcal{R}_{T_2}(a, \widehat{M}_{T_1}) - \mathcal{R}(a, \widehat{M}_{T_1})\right| \geq \varepsilon\right\}$$

$$\leq \mathcal{N}_{d_\mathcal{A}}(\delta, \mathcal{A}) \max_{\mathcal{N}_{d_\mathcal{A}}(\delta,\mathcal{A})}\left(16\mathcal{N}_1\left(\frac{\varepsilon - K\alpha\delta}{16}, L\circ\mathcal{A}_i^\delta, T_2\right)\exp\left(-\frac{(\varepsilon - K\alpha\delta)^2\left[\frac{T_2^{\frac{s}{1+s}}}{2}\right]}{128K^2}\right) + \beta_0\left[T_2^{\frac{s-\gamma}{1+s}}\right]\right)$$

Error for Mechanism Learning

GSP with d-dim linear reserve price function: $16\mathcal{N}_1\left(\frac{\varepsilon}{16}, Rev\circ\mathcal{A}_i^\delta, T\right) \leq \left(\frac{eT_2K}{\epsilon}\right)^{16|\mathcal{B}|d}$

Generalization Analysis

$$\min_{B \in \mathcal{B}} l(B; a_0, u_1, \dots, u_{T_1}) \triangleq B_{T_1}$$

$$\Downarrow$$

$$\min_{a \in \mathcal{A}} L(a; B_{T_2}; u_1, \dots, u_{T_2}) \triangleq a_{T_2}$$

$$\Downarrow$$

$$R(a_{T_2}, \mathrm{B}^*) = \lim_{T_3 \to \infty} E_{P(u)} L(a_{T_2}, \mathrm{B}^*, u_1, \dots, u_{T_3})$$

Objective: $\lim_{T_1, T_2 \to \infty} R(a_{T_2}, B^*) \xrightarrow{p} R^{OPT}$

Error Decomposition

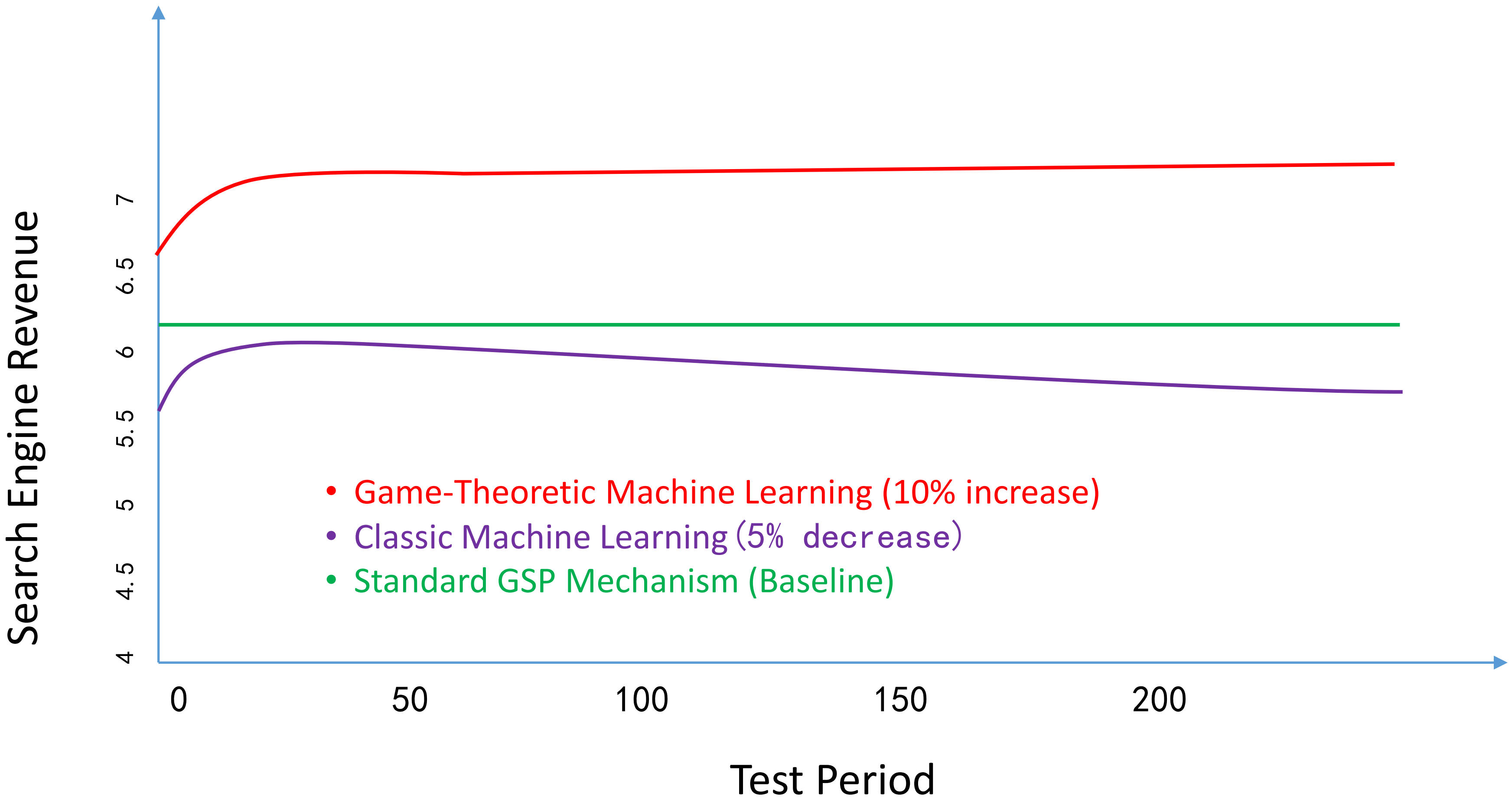
$$R(a_{T_2}, B^*) - R(a^*, B^*) = 2\mathrm{K}\sup_{a \in \mathcal{A}} C(a, B^*) \boxed{\left\| B_{T_1} - B^* \right\|_\infty} + 2 \sup_{a \in \mathcal{A}} \boxed{\left| R(a_{T_2}, B_{T_1}) - R(a, B_{T_1}) \right|}$$

Parametric Method: $\mathrm{P}\left\{ \left\| B_{T_1} - B^* \right\|_\infty \geq \epsilon \right\} \leq 2e^{-\frac{\left(T_1 \epsilon |\mathcal{B}|^2 |\mathcal{H}| \delta_0 - 2C_1 N_0\right)^2}{2T_1 N_0^2 C_1^2}}$

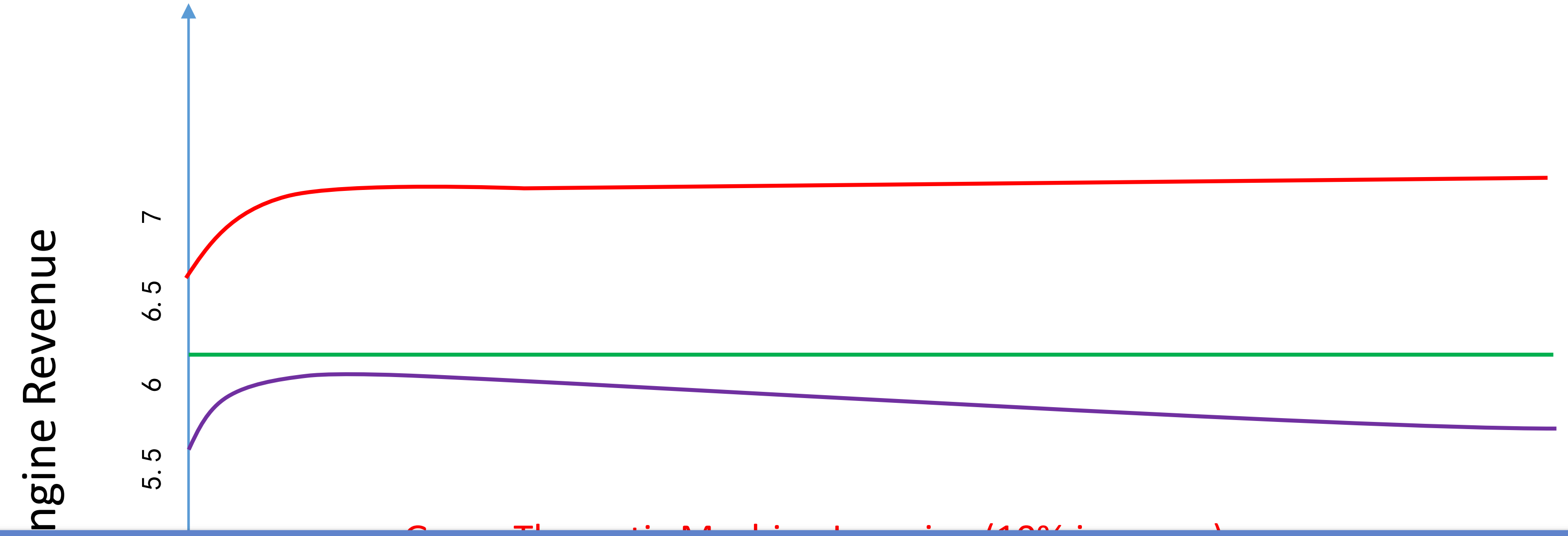
Error for Behavior Learning

- The overall error bound will converge to zero when the scales of agent behavior data T_1 and user data T_2 approach infinity;
- The convergence rate w.r.t. T_1 is faster than that w.r.t. T_2, indicating that one needs more user data than agent behavior data for training.

Experimental Results



Experimental Results



Beyond empirical success: Is there theoretic guarantee?
Can game-theoretic learning generalize? What kind of training data is desirable for game theoretic learning?

Summary of Part II

- Machine learning could be applied to user click behavior modeling, advertising bidding behavior modeling, and auction mechanism optimization.
- For user click behavior modeling, one needs to consider appropriate feature engineering, externality between ads, and temporal dependency between sequential clicks.
- Given the existence of strategic advertisers, the data distribution of bidding data will change in response to auction mechanism change. Game theoretic machine learning was proposed to handle this complex case, whose generalization ability and empirical performance were examined.

Future Research Directions

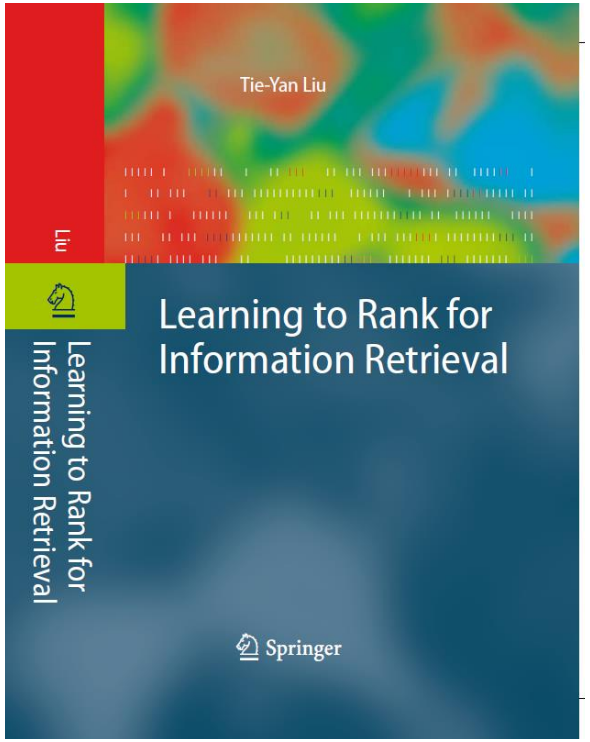
Machine Learning for Search Ranking

- Learning to rank
  - Online learning to rank (contextual bandits, etc.)
  - Structural learning to rank (diversity, whole-page relevance)
  - Large scale learning to rank (parallelization, effective sampling)
  - …
- Other topics
  - Query suggestion
  - User modeling
  - Question answering
  - …

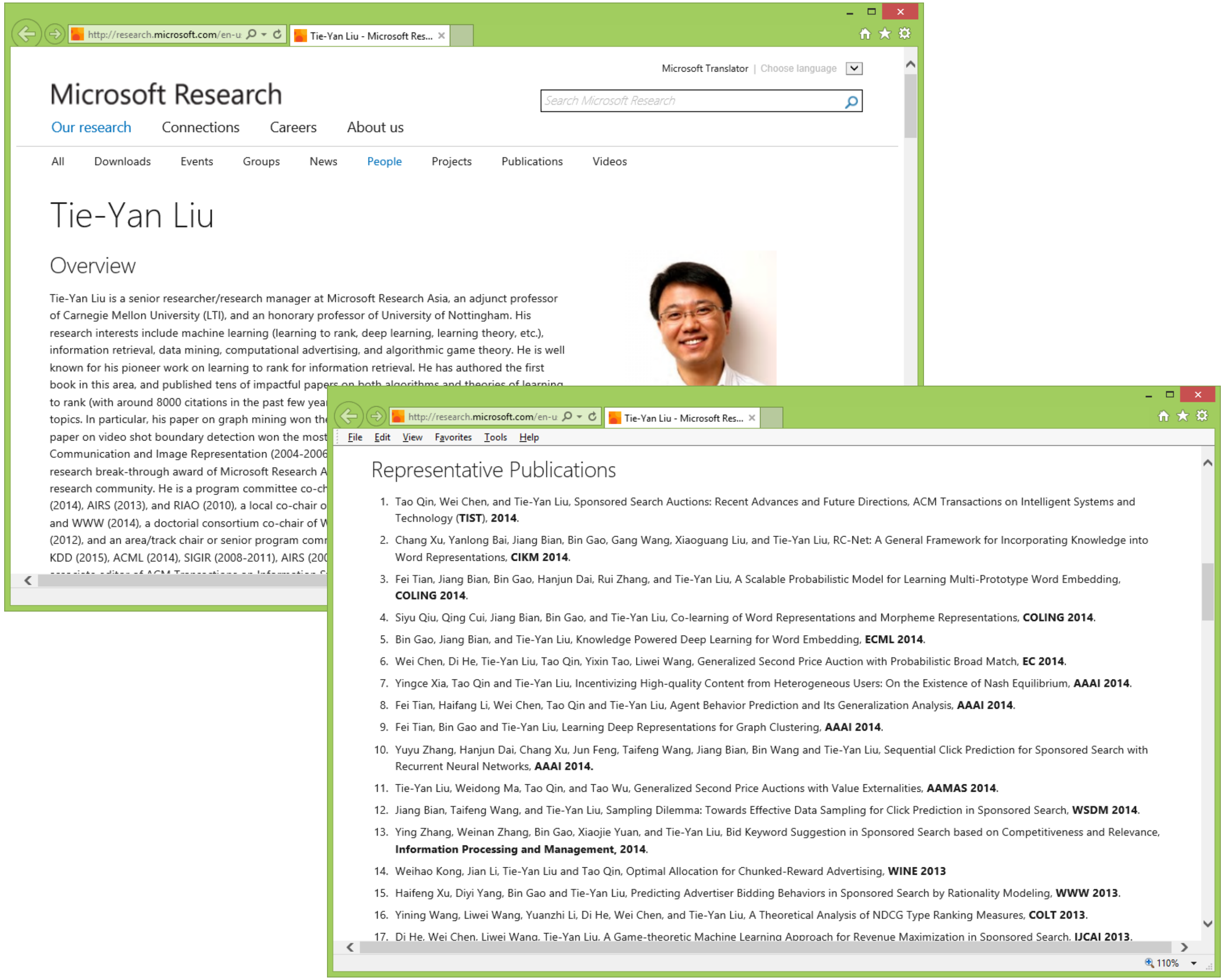
Machine Learning for Ad Auctions

- Game theoretic learning
    - Efficient learning algorithm (surrogate loss, optimization methods)
    - Online game-theoretic learning (Markov bandits, etc.)
    - Applications to new domains (recommender systems, social networks, crowdsourcing, mobile apps, etc.)
    - …
- Other topics
    - Large scale learning for click prediction
    - Personalization
    - Rich search ads, display ads, social ads, mobile ads
    - …

References



Tie-Yan Liu
Liu
Learning to Rank for Information Retrieval
Learning to Rank for Information Retrieval
Springer



http://research.microsoft.com/en-u
Tie-Yan Liu - Microsoft Res...
Microsoft Translator | Choose language
Microsoft Research
Search Microsoft Research
Our research
Connections
Careers
About us
All
Downloads
Events
Groups
News
People
Projects
Publications
Videos
Tie-Yan Liu
Overview
Tie-Yan Liu is a senior researcher/research manager at Microsoft Research Asia, an adjunct professor
of Carnegie Mellon University (LTI), and an honorary professor of University of Nottingham. His
research interests include machine learning (learning to rank, deep learning, learning theory, etc.),
information retrieval, data mining, computational advertising, and algorithmic game theory. He is well
known for his pioneer work on learning to rank for information retrieval. He has authored the first
File Edit View Favorites Tools Help
Representative Publications
1. Tao Qin, Wei Chen, and Tie-Yan Liu, Sponsored Search Auctions: Recent Advances and Future Directions, ACM Transactions on Intelligent Systems and Technology (TIST), 2014.
2. Chang Xu, Yanlong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu, RC-Net: A General Framework for Incorporating Knowledge into Word Representations, CIKM 2014.
3. Fei Tian, Jiang Bian, Bin Gao, Hanjun Dai, Rui Zhang, and Tie-Yan Liu, A Scalable Probabilistic Model for Learning Multi-Prototype Word Embedding, COLING 2014.
4. Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu, Co-learning of Word Representations and Morpheme Representations, COLING 2014.
5. Bin Gao, Jiang Bian, and Tie-Yan Liu, Knowledge Powered Deep Learning for Word Embedding, ECML 2014.
6. Wei Chen, Di He, Tie-Yan Liu, Tao Qin, Yixin Tao, Liwei Wang, Generalized Second Price Auction with Probabilistic Broad Match, EC 2014.
7. Yingce Xia, Tao Qin and Tie-Yan Liu, Incentivizing High-quality Content from Heterogeneous Users: On the Existence of Nash Equilibrium, AAAI 2014.
8. Fei Tian, Haifang Li, Wei Chen, Tao Qin and Tie-Yan Liu, Agent Behavior Prediction and Its Generalization Analysis, AAAI 2014.
9. Fei Tian, Bin Gao and Tie-Yan Liu, Learning Deep Representations for Graph Clustering, AAAI 2014.
10. Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang and Tie-Yan Liu, Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks, AAAI 2014.
11. Tie-Yan Liu, Weidong Ma, Tao Qin, and Tao Wu, Generalized Second Price Auctions with Value Externalities, AAMAS 2014.
12. Jiang Bian, Taifeng Wang, and Tie-Yan Liu, Sampling Dilemma: Towards Effective Data Sampling for Click Prediction in Sponsored Search, WSDM 2014.
13. Ying Zhang, Weinan Zhang, Bin Gao, Xiaojie Yuan, and Tie-Yan Liu, Bid Keyword Suggestion in Sponsored Search based on Competitiveness and Relevance, Information Processing and Management, 2014.
14. Weihao Kong, Jian Li, Tie-Yan Liu and Tao Qin, Optimal Allocation for Chunked-Reward Advertising, WINE 2013
15. Haifeng Xu, Diyi Yang, Bin Gao and Tie-Yan Liu, Predicting Advertiser Bidding Behaviors in Sponsored Search by Rationality Modeling, WWW 2013.
16. Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu, A Theoretical Analysis of NDCG Type Ranking Measures, COLT 2013.
17. Di He, Wei Chen, Liwei Wang, Tie-Yan Liu, A Game-theoretic Machine Learning Approach for Revenue Maximization in Sponsored Search, IJCAI 2013.
110%

Thanks

tyliu@microsoft.com
http://research.microsoft.com/users/tyliu/
http://weibo.com/tieyanliu/