Machine Learning for Search Ranking and Ad Auctions

Tie-Yan Liu
Senior Researcher / Research Manager
Microsoft Research
Microsoft Research Asia

- Founded on Nov. 5th, 1998
- 4000+ papers published on top-tier journals/conferences, 50+ best paper awards
- 360+ technologies transferred to Microsoft products
- Diverse research topics: ML, IR, MM, CV, HW, HCI, SYS, Theory, Econ

Redmond, Washington, USA Sep 1991
Cambridge, Massachusetts, USA 2008
New York, USA May 2012
Cambridge, UK July 1997
Bangalore, India Jan 2005
Beijing, China Nov 1998
Search and Ads
Eco-system

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

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

Search engines
- 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
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

Advertising revenue market share by media - 2013 ($ billions)

- Internet: $42.8
- Broadcast Television: $40.1
- Cable Television: $34.4
- Newspaper: $18.0
- Radio: $16.7
- Magazine (Consumer): $13.4
- Out of Home: $7.9
- Video Game: $0.9
- Cinema: $0.8

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

Search – Ranking

• Heuristically designed ranking models

\[ f_{BM25}(D, Q) = \sum_{i} \frac{IDF(q_i)f(q_i,D)(k_1+1)}{f(q_i,D)+k_1(1-b+b_{AVDL})}, \]
\[ IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \]

• \( PRank(v_i) = \sum_{v_j \in inlink[v_i]} \frac{PRank(v_j)}{|outlink[v_j]|} \)

Ads – Auction

• Heuristically designed ranking and pricing rules

• GSP auction ranks ads by \( p\text{Click} \times bid \)

• GSP auction charges clicked ads by \( \frac{p\text{Click}_{next}\times\text{bid}_{next}}{p\text{Click}} \).

• 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
  - \[ f_{BM25}(D, Q) = \sum_i \frac{IDF(q_i)f(q_i,D)(k_1+1)}{f(q_i,D)+k_1(1-b+b\frac{|D|}{AVDL})}, \]
  - \[ IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{0.5} \]

**Ads – Auction**
- Heuristically designed ranking and pricing rules
  - GSP auction ranks ads by \( p_{Click} \times bid \)
  - GSP auction charges clicked ads by \( \frac{p_{Click_{next}} \times bid_{next}}{p_{Click}} \)

• 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?
Instead of relying on heuristics and assumptions, we can let the data speak for us.
Learning to Rank
-- Machine Learning for Search Ranking
A New Problem?

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].

3. **Reduce ranking to pairwise classification**
   - Classify the order between each pair of documents.
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 $f : f(A) = 3, f(B) = 0, f(C) = 1$ \quad ACB
  - Model $h : h(A) = 4, h(B) = 6, h(C) = 3$ \quad BAC
  - ground truth $g : g(A) = 6, g(B) = 4, g(C) = 3$ \quad ABC

• According to NDCG or AP: $\text{sim}(f, g) > \text{sim}(g, h)$

• Pointwise distance/pairwise comparison contradicts ranking measures
  - According to pointwise distance: $\text{sim}(f, g) < \text{sim}(g, h)$.
  - According to pairwise comparison: $\text{sim}(f, g) = \text{sim}(g, h)$. 
Listwise Approach

• Instead of considering individual documents or document pairs, treat the entire document set $x^q = (x^q_1, ..., x^q_M)$ 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 ↔ Permutation probability distribution

\[ P(\pi | f) = \prod_{j=1}^{m} \frac{\exp(f(x_{\pi(j)}))}{\sum_{k=j}^{m} \exp(f(x_{\pi(k)}))} \]

\[ P_{PL}(ABC | f) = \frac{\exp(f(A))}{\exp(f(A)) + \exp(f(B)) + \exp(f(A))} \cdot \frac{\exp(f(B))}{\exp(f(B)) + \exp(f(C))} \cdot \frac{\exp(f(C))}{\exp(f(C))} \]

- \( P(A \text{ ranked No.1}) \)
- \( P(B \text{ ranked No.2} | A \text{ ranked No.1}) \)
- \( P(C \text{ ranked No.3} | A \text{ ranked No.1}, B \text{ ranked No.2}) \)
Distance between Ranked Lists

- **f**: \( f(A) = 3, f(B) = 0, f(C) = 1 \);
  Ranking by \( f \): ABC

- **g**: \( g(A) = 6, g(B) = 4, g(C) = 3 \);
  Ranking by \( g \): ABC

- **h**: \( h(A) = 4, h(B) = 6, h(C) = 3 \);
  Ranking by \( h \): ACB

Distance measures:

- \( d(f, g) = 0.46 \)
- \( d(g, h) = 2.56 \)

Using **KL-divergence** to measure difference between distributions.
Listwise Loss Functions

• **ListNet** [Learning to Rank: From Pairwise Approach to Listwise Approach. ICML 2007]
  • Minimize KL divergence between permutation probabilities of ranking model and ground truth
  • \( L(f; x, g) = D(P_{PL}(\pi|g)||P_{PL}(\pi|f(x))) \)

• **ListMLE** [Listwise Approach to Learning to Rank: Theorem and Algorithm, ICML 2008]
  • Directly maximize likelihood of ground truth permutation induced by ranking model
  • \( P_{PL}(\pi|g) \rightarrow P_g(\pi) = \begin{cases} 1, & \pi = \pi_g \\ 0, & \text{otherwise} \end{cases} \rightarrow L(f; x, g) = -\log P_{PL}(\pi_g|f(x)) \)

• Model general ground truth labels using “equivalent permutation set”:
  • \( L(f; x, g) = \min_{\pi \in \Omega_g} (-\log P_{PL}(\pi|f(x))) \)
  • For relevance degree or clicks: \( \Omega_g = \{\pi|u < v, if g(\pi^{-1}(u)) > (\pi^{-1}(v))\} \)
  • For pairwise preference: \( \Omega_g = \{\pi|u < v, if g(\pi^{-1}(u), \pi^{-1}(v)) = 1\} \)
Experimental Results on LETOR Benchmark

Winning number: comparison with other algorithms over all 7 sub datasets in LETP0R

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

http://research.microsoft.com/~letor/
Listwise Ranking Functions

• Beyond pointwise ranking function
  • Instead of simply sorting the scores assigned to individual documents, also consider relationship between documents (topic diversity, domain hierarchy).

• C-CRF [Global Ranking Using Continuous Conditional Random Fields, NIPS 2008]

\[
P(g^q | x^q) = \frac{1}{Z(x^q)} \exp \{ \sum_i \sum_k \alpha_k h^1_k(g^q_i, x^q) + \sum_{i,j} \sum_k \beta_k h^2_k(g^q_i, g^q_j, x^q) \}
\]

• R-RSVM [Learning to Rank Relational Objects and Its Application to Web Search, WWW 2008]

\[
f(x) = \arg \min_z \{ \alpha l_1(h(x; w), z) + \beta l_2(R, z) \}
\]
Algorithmic development of learning to rank is booming
Generalization Analysis for Learning to Rank

Loss $L(f; x, g)$ on finite data

Training Process  \rightarrow  Ranking Model  \rightarrow  Test Process

Measure (1-NDCG, 1-MAP) on infinite data

Can this process generalize?

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

Structured training data

Queries

Web Documents

New theory is needed due to uniqueness of ranking
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”.

[Graph showing the relationship between training set size and test set NDCG.]

[Yilmaz and Robertson, 2009]
Previous Assumptions

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

```
Documents

Doc 1  Doc 2  Doc 3  Doc m
Label 1 Label 2 Label 3 Label m
```

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

```
Queries

query 1  ......  query n

Doc Set 1  Doc Set n
Label Set 1  Label Set n
```

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

2014/9/18
Two-layer Sampling

[Two-Layer Generalization Analysis for Ranking Using Rademacher Average, NIPS 2010]
Two-layer Sampling

[Two-Layer Generalization Analysis for Ranking Using Rademacher Average, 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 $l^o\mathcal{F}$ is bounded by $M$, and Rademacher average $E[\mathcal{R}_m(l^o\mathcal{F})]$ is bounded by $D(l^o\mathcal{F}, m)$, then for arbitrary sample distribution, for $\forall f \in \mathcal{F}$, with probability at least $1 - \delta$, $R^l(f) \leq \hat{R}^l_{m_1, \ldots, m_n}(f) + D(l^o\mathcal{F}, n) + \sqrt{\frac{2M^2 \log(\frac{M}{\delta})}{n}} + \frac{1}{n} \sum_i D(l^o\mathcal{F}, \left\lceil \frac{m_i}{2} \right\rceil) + \sum_i \frac{2M^2 \log(\frac{M}{\delta})}{m_i n^2}$.

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:

$$n^* = \frac{c_1 \sqrt{V} + \sqrt{2\log(\frac{4}{\delta})}}{c_1 \sqrt{2V}} \sqrt{C}; \quad m_i^* \equiv \frac{C}{n^*}$$
Theorem 1. Suppose $l$ is the loss function for ranking, $\mathcal{F}$ is the function class of the ranking model, and $l^o \mathcal{F}$ is bounded by $M$, and Rademacher average $E[\mathcal{R}_m(l^o \mathcal{F})]$ is bounded by $D(l^o \mathcal{F}, m)$, then for arbitrary sample distribution, for $\forall f \in \mathcal{F}$, with probability at least $1 - \delta$,

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

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

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

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 ≤ Training Loss + \( \varepsilon(n,m,F) \)
Loss Function vs. Ranking Measure

• Loss Function in ListMLE, as an example

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

Based on the scores \( f(x) \) 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 \( \pi \) 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

[Ranking Measures and Loss Functions in Learning to Rank, NIPS 2009]

- Model ranking as a sequence of classifications

  Ground truth permutation: \( g = \{A > B > C > D\} \)

  Prediction of the ranking function \( f \): \( \pi = \{B > A > D > C\} \)

\[
\begin{align*}
&g = \{A, B, C, D\} \\
&\pi = \{B, A, D, C\} \\
&T_f(x_{(s)}) \text{ is the classifier taking the document } x_{(s)} \\
&\text{ as input and output the document with the largest ranking score.}
\end{align*}
\]

The weighted classification error for each step in the sequence

\[
L_\beta(f; x, g) = \sum_s \beta(s) I_{\{T_f(x_{(s)}) \neq g(s)\}}
\]
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) $1 - NDCG(f; x, g) \leq Z_n L_{\beta_1}(f; x, g)$, where $\beta_1(s) = \frac{c(g(\pi^{-1}(s)))}{\log(s+1)}$.

2) $1 - AP(f; x, g) \leq \frac{1}{\sum_{i\geq k} n_i} L_{\beta_2}(f; x, g)$, where $\beta_2(s) \equiv 1$.

**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 $\forall f$:

$$L_{\beta}(f; x, g) \leq \frac{\max \beta(s)}{\ln 2} L(f; x, g)$$

$$(1-NDCG), (1-MAP) \leq \text{Essential Loss} \leq \text{Pairwise and Listwise Loss Functions}$$
Game-theoretic Learning
-- Machine Learning for Ad Auction
Learning to Rank for Auction Optimization?

• RankLogistic [Zhu et al. SIGIR 2009]
  • Training data: historical auction logs (queries, clicks, bids)
  • Objective function: empirical revenue on historical data
  • Hypothesis space: ranking function + first price rule

\[
R(w) = \sum_{q \in Q} \sum_{p=1}^{n_q} rev_q(p) c_q(p) \mathbb{I}\left\{\min_{i \neq p}\{f(w,x_q(p)) - f(w,x_q(i))\} > 0\right\}
\]
What’s the Problem?

- Learning to rank

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

- Game-theoretic learning

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

[A Game-theoretic Machine Learning Approach for Revenue Maximization in Sponsored Search, IJCAI 2013]

Learn from historical behavior data how an advertiser behave in certain situations

**Historical Behavior Data**

**Behavior Learner**

Learn from historical behavior data how an advertiser behave in certain situations

**Agent Behavior Model**

\[
\min_{B \in \mathcal{B}} l(B; a_0, u_1, \ldots, u_{T_1}) \triangleq B_{T_1}
\]

**Historical User Data**

**Mechanism Learner**

Predict advertisers’ bidding behaviors using the behavior model, and optimize the mechanism on predicted behaviors

**Optimal Mechanism**

\[
\min_{a \in \mathcal{A}} L(a; B_{T_2}, u_1, \ldots, u_{T_2}) \triangleq a_{T_2}
\]
How to Learn Advertiser Behaviors?

**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
  [Agent Behavior Prediction and Its Generalization Analysis, 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 $b_{t+1}$:
  \[ P(b_{t+1}|all\ info) = P(b_{t+1}|I_t, b_t) = \prod_{i=1}^{n} P_i(b^i_{t+1}|I^i_t, b^i_t). \]

*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

<table>
<thead>
<tr>
<th>Model</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best response model</td>
<td>$P(b_{t+1}^i</td>
</tr>
<tr>
<td>Quantal response model</td>
<td>$P(b_{t+1}^i</td>
</tr>
<tr>
<td>Random model</td>
<td>$P(b_{t+1}^i</td>
</tr>
<tr>
<td>I.I.D. model</td>
<td>$P(b_{t+1}^i</td>
</tr>
</tbody>
</table>
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.

\[ P(b_{t+1}^i | I_t, b_t^i) = \mathbb{I}\{b_{t+1} = \arg \max_b U_i^i(I_t, b)\} \]

→ Nash Equilibrium

\[ P(b_{t+1}^i | I_t, b_t^i) \propto U_i^i(I_t, b_{t+1}^i) \]

→ 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:
  - \( B_{T_1} = \arg\min_{B \in \mathcal{B}} l(B; a_0, u_1, ..., u_{T_1}) \)

**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:
  - \( \hat{P}_i(b'|I, b) = \frac{\sum_{t=1}^{T_1} \mathbb{I}_{\{b_{t+1}=b'|b_t=b,h_t=H_j\}}}{\sum_{t=1}^{T_1} \mathbb{I}_{\{b_t=b,h_t=H_j\}}} \)
Accuracy

Advertiser Behavior Prediction in Real Data

Error Rate

Days in bidding logs
Game-Theoretic Learning

1: Auction Implement

- Implement above process on user data with size $T_2$, and compute the loss function (i.e., minus average search engine revenue)
  \[
  L(a; B_{T_1}; u_1, ..., u_{T_2}) = -\frac{1}{T_2} \sum_{t=1}^{T_2} Rev(a; b_t(B_{T_1}), u_t).
  \]

2: Bid Update

- Learn optimal auction mechanism by minimizing the loss function
  \[
  a_{T_2} = \arg\min_{a \in \mathcal{A}} L(a; B_{T_1}; u_1, ..., u_{T_2}), \text{ where } \mathcal{A} \text{ is the auction mechanism space}
  \]

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: $Rev(a; b_t, u_t)$.

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 $t + 1$. 
Experimental Results

- Game-Theoretic Machine Learning (10% increase)
- Classic Machine Learning (5% decrease)
- Standard GSP Mechanism (Baseline)
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?
Generalization Analysis

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

Error Decomposition

\[
R(a_{T_2}, B^*) - R(a^*, B^*) = 2K \sup_{a \in A} (a, B^*) ||B_{T_1} - B^*||_a + 2 \sup_{a \in A} R(a_{T_2}, B_{T_1}) - R(a, B_{T_1})
\]

Parametric Method:

\[
P \left\{ \|B_{T_1} - B^*\|_\infty \geq \varepsilon \right\} \leq 2e \frac{-\left(\frac{\varepsilon}{2} \frac{\|B\|_\infty^2}{2 T_1 N_0^2 C_1^2} - \left(\frac{\varepsilon}{2} \frac{\|B\|_\infty^2}{2 T_1 N_0^2 (|\|B\|_\infty + 1)^2} \right)\right)}{2T_1 N_0^2 (|\|B\|_\infty + 1)^2}
\]

Non-Parametric Method:

\[
P \left\{ \|B_{T_1} - B^*\|_\infty \geq \varepsilon \right\} \leq 2Ce \frac{-\left(\varepsilon - Ka\delta \right)^2}{2T_1 N_0^2 (|\|B\|_\infty + 1)^2} \left( \exp \left( \frac{(\varepsilon - Ka\delta)^2}{128K^2} \right) \right) + \beta_0 \left( \frac{\varepsilon T_2}{T_2^2} \right)
\]

GSP with \(d\)-dim linear reserve price function:

\[
16N_1 \left( \frac{\varepsilon}{16}, \text{Rev} \circ \mathcal{A}_T^\delta, T \right) \leq \left( \frac{e T_2 K}{\varepsilon} \right)^{16|\|B\|_d|}
\]

Error for Behavior Learning

Error for Mechanism Learning
Generalization Analysis

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

Error Decomposition

\[ R(a_{T_2}, B^*) - R(a^*, B^*) = 2K \sup_{a \in \mathcal{A}} C(a, B^*) \left| B_{T_1} - B^* \right|_2 + 2 \sup_{a \in \mathcal{A}} \left| R(a_{T_2}, B_{T_1}) - R(a, B_{T_1}) \right| \]

Parametric Method: \( P \left\{ \left| B_{T_1} - B^* \right|_2 \geq \epsilon \right\} \leq 2e^{-\frac{\left( 2T_1 \epsilon^2 \left| B \right|_2 \left| B_0 - 2C_1 N_0 \right| \right)^2}{2T_1 N_0 C_1^2}} \) (for \( \epsilon \to 0 \))

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.
Future Directions

• 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)

• 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.)
Thanks

tyliu@microsoft.com
http://research.Microsoft.com/users/tyliu/

Special thanks to my colleagues: Tao Qin and Wei Chen