Online Search and Rank Optimization

May 2019

Will Wang

joint with Giovanni Compiani (Berkeley), Greg Lewis (Microsoft), Sida Peng (Microsoft)
Online search is ubiquitous
How to design online search platforms?

• Focus on search behavior and product rankings
• Long-run relationships with consumers: maximizing consumer surplus

• Understand search behavior:
  • Rich data in search patterns and choices
  • A general framework that subsumes several existing models (e.g. Weitzman 1979)

• Ranking optimization (under parametric assumptions):
  • Ranking affects salience and thus search patterns
  • Computationally tractable
Literature


How do consumers search?

- A general double index framework (subsuming several popular search models)

- Each product has a search and utility index \((s_j, u_j)\)
- The consumer knows the search indices and the utility of the outside option \(u_0\)
- The remaining utilities \(\{u_j\}_{j=1}^J\) are unknown
- Consumer learns \(u_j\) by searching for product \(j\)

- Rank goods according to \(s\): \(s_1 \geq s_2 \geq \cdots \geq s_J\)
- If \(s_1 < u_0\), stop searching without purchase
- Else, search good 1, then
  - If \(s_2 < \max\{u_0, u_1\}\), stop searching and maximize utility over goods 0 and 1
  - Else, search good 2, then
    - ...
Why not rank by “product relevance”? 

• How to best rank the alternatives to maximize consumer surplus?

• Consider an example: 3 products Red, Green, and Blue; $u_0 = 0$
  • Ranking affects search index on top of a “baseline” search index (based on other attributes)

• Further complicated with consumer heterogeneity

(s, u)

Rank: 1
(2,5)

Rank: 2
(2,4)

Rank: 3
(5,3)

(s, u)

(4,4)
(0,5)
(5,3)
The product that gets purchased is the one with the highest \( \min \{ u_{ij}, s_{ij} \} \) (modifying Choi, Dai, Kim, ECMA 2018) – i.e. both salience and utility matter for choice.

Introduce “double logit”: Let \( s_{ij} = \delta_j^S + \epsilon_{ij}^S \) and \( u_{ij} = \delta_j^U + \epsilon_{ij}^U \), where \( \epsilon \)'s are distributed Type I EV.

Assume perfect correlation of errors: \( \min \{ u_{ij}, s_{ij} \} = v_{ij} = \delta_j^V + \epsilon_{ij} \), where \( \delta_j^V = \min \{ \delta_j^S, \delta_j^U \} \).

Ex-ante consumer surplus (i.e. prior to the realization of the common logit shock):

\[
E(CS) = \log \left( \sum_j \exp \delta_j^V \right) + \sum_{j: \delta_j^U > \delta_j^S} \frac{\exp \delta_j^V}{\sum_k \exp \delta_k^V (\delta_j^U - \delta_j^S)}
\]
Rank optimization

• Allocate resource $r$ from budget $R$ to increase salience: \( \delta_j^S = \delta_{j0}^S + r \)

• Marginal benefit of rank promotion: let \( q_j = \frac{\exp \delta_j^V}{\sum_k \exp \delta_k^V} \), when \( \delta_j^U > \delta_j^S \),

\[
\frac{\partial (E(CS))}{\partial r} = q_j (\delta_j^U - \delta_j^S) - q_j \sum_k q_k (\delta_k^U - \delta_k^S)
\]

• “Potential” (=\( \delta ^U - \delta ^S \)): Always only promote positive potential products

• Salience: Within the set of positive potential products, baseline market share matters

• Avoid two mistakes:
  • Don’t promote products that will be ultimately disappointing
  • Don’t promote products that have high potential but very low baseline salience
Rank optimization

• Allocate resource $r$ from budget $R$ to increase salience: $\delta^S_j = \delta^S_{j0} + r$

• Marginal benefit of rank promotion: let $q_j = \frac{\exp \delta^V_j}{\sum_k \exp \delta^V_k}$, when $\delta^U_j > \delta^S_j$

$$\frac{\partial (E(CS))}{\partial r} = q_j (\delta^U_j - \delta^S_j) - q_j \sum_k q_k (\delta^U_k - \delta^S_k)$$

• “Potential” ($=\delta^U - \delta^S$): Always only promote positive potential products

• Salience: Within the set of positive potential products, baseline market share matters

• Avoid two mistakes:
  • Don’t promote products that will be ultimately disappointing
  • Don’t promote products that have high potential but very low baseline salience
A greedy algorithm

• Discrete ranks require an algorithm

• For each rank $r = 1, 2, \ldots, J$, let $\delta_j^S = \delta_{j0}^S + f(r)$ if product $j$ is ranked at $r$
  
  • If only one or no products with potential $(\delta_j^U - \delta_{j0}^S)$ greater than $f(r)$
    
    $\Rightarrow$ rank product $j$ with highest potential at rank $r$ ("Minimize waste")

  • If multiple products with potential greater than $f(r)$
    
    $\Rightarrow$ rank product $j$ with highest $\frac{\partial (E(CS))}{\partial r}$ among products with potential greater than $f(r)$
Simulations

• Base case: $\delta^U, \delta^S \sim i.i.d. N(0,1)$
  - Ranking technology: $f(r) = A \cdot \exp(-r)$, where $A=5$

• Normalize CS from worst result (0) to best result (1) across all possible rankings

• Results hold for many products
  - Our algorithm is fast to compute when enumeration is infeasible

<table>
<thead>
<tr>
<th>Enumerating all rankings</th>
<th>Base</th>
<th>Strong ranking (A=15)</th>
<th>High utility / Weak ranking (\bar{u}=3; A=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best result</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Average result</td>
<td>30%</td>
<td>35%</td>
<td>58%</td>
</tr>
<tr>
<td>Worst result</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ranking by</th>
<th>Base</th>
<th>Strong ranking (A=15)</th>
<th>High utility / Weak ranking (\bar{u}=3; A=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our algorithm</strong></td>
<td>95%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>Utility</td>
<td>89%</td>
<td>93%</td>
<td>93%</td>
</tr>
<tr>
<td>Potential: utility - base search index</td>
<td>96%</td>
<td>98%</td>
<td>88%</td>
</tr>
</tbody>
</table>
Empirical application: Microsoft Partner Center

• Launched in 2017 to help match Microsoft partners among themselves to exchange goods and services
• The Partner Center generates roughly 1000 matches per day with over 8000 global partners

• Our data contains 7638 unique customer visits and 14066 clicks on partner profiles (searches)
  • Half of the customers chose at least one partner (i.e. submitting a referral)
    • 42% of the customers search before choosing
    • 7% of the customers directly chose partners
  • The other half of the customers did not choose any partner:
    • 36% of the customers left after some searching
    • 16% of the customers left without any search
Microsoft Partner Center

Connect with trusted Microsoft solution providers
Microsoft certified solution providers specialize in providing up-to-date Microsoft technology-based customer solutions all over the world.

Save time and money
A certified Microsoft solution provider can assess your business goals, identify a solution that meets your business needs and help your business become more agile and efficient.

Grow your business
Microsoft solution providers can help you take full advantage of the cloud, opening a wide array of new opportunities for you to grow your business and your revenue.
Connect with trusted Microsoft solution providers
Microsoft certified solution providers specialize in providing up-to-date Microsoft technology-based customer solutions all over the world.

Save time and money
A certified Microsoft solution provider can assess your business goals, identify a solution that meets your business needs and help your business become more agile and efficient.

Grow your business
Microsoft solution providers can help you take full advantage of the cloud, opening a wide array of new opportunities for you to grow your business and your revenue.
Microsoft Partner Center

I'm looking for a solution provider...

Connect with trusted Microsoft solution providers
Microsoft certified solution providers specialize in providing up-to-date Microsoft technology-based customer solutions all over the world.

Save time and money
A certified Microsoft solution provider can assess your business goals, identify a solution that meets your business needs and help your business become more agile and efficient.

Grow your business
Microsoft solution providers can help you take full advantage of the cloud, opening a wide array of new opportunities for you to grow your business and your revenue.
Microsoft Partner Center

Logicworks

This provider has demonstrated competency in the following areas:

- **Gold**
  - Cloud Platform

- **Silver**
  - Cloud Platform

Microsoft endorses this provider for the following skills and capabilities:

- Azure, Enterprise Mobility + Security, Office 365, SQL, Windows Consulting and Professional, Custom Solution, Deployment or Migration, Integration, Licensing, Managed Services (MSP), Project Services, Backup & Disaster Recovery, Cloud Database Migration, Cloud Migration, Containers, DevOps, Hybrid Storage, Networking, Serverless Computing, SQL Server Upgrade

About Us:

Logicworks is a Gold Cloud Platform Partner and Managed Services Partner that helps customers plan, build, and run Azure cloud infrastructure. For companies that are new to Azure, Logicworks can help develop a cloud strategy, conduct a TCO analysis, and architect a secure cloud environment. Companies with existing Azure environments can leverage Logicworks for 24x7 technical support and access to a proprietary suite of monitoring and automation tools.

Top Locations:

- **155 Avenue of the Americas, 8th Floor, New York, NY, US 10013**
- **1550 Wewatta Street, Denver, CO, US 80202**
- **500 Yale Avenue, North, Seattle, WA, US 98109**
- **2401 Walnut Street, Suite 100, Philadelphia, PA, US 19103**
- **745 Atlantic Avenue, Boston, MA, US 02111**
Empirical specification

• We formulate the double index model into a discrete choice problem

• Define a consumer $i$’s utility and search index for product $j$ as

$$u_{ij} = x_j \beta + \omega_j \gamma + \xi_j + v_j + \epsilon^U_{ij}$$

$$s_{ij} = x_j \beta^S + position_j \gamma^S + \xi_j^S + \epsilon^S_{ij}$$

where

• $x_j$’s are observed characteristics before search, e.g. endorsement badge, LinkedIn logo
• $\omega_j$’s are observed characteristics after search, e.g. endorsement details, partner attributes
• $position_j$ include both the vertical groups and horizontal rankings of product $j$

• $\xi_j$’s are unobserved characteristics before search
• $v_j$’s are unobserved characteristics after search
• $\xi_j^S$’s are unobserved search index
• $\epsilon$’s are i.i.d. distributed Type I EV
Estimates

<table>
<thead>
<tr>
<th></th>
<th>Search score</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coef</td>
<td>SE</td>
</tr>
<tr>
<td>Group 1</td>
<td>0.5344***</td>
<td>0.0995</td>
</tr>
<tr>
<td>Group 2</td>
<td>-1.2401***</td>
<td>0.1086</td>
</tr>
<tr>
<td>Rank 1</td>
<td>3.5620***</td>
<td>0.1370</td>
</tr>
<tr>
<td>Rank 2</td>
<td>2.4517***</td>
<td>0.1296</td>
</tr>
<tr>
<td>Rank 3</td>
<td>2.1400***</td>
<td>0.1346</td>
</tr>
<tr>
<td>Rank 4</td>
<td>1.7604***</td>
<td>0.1448</td>
</tr>
<tr>
<td>Rank 5</td>
<td>1.6875***</td>
<td>0.2024</td>
</tr>
<tr>
<td>Rank 6</td>
<td>1.6440***</td>
<td>0.1489</td>
</tr>
<tr>
<td>Rank 7</td>
<td>0.5501*</td>
<td>0.1904</td>
</tr>
<tr>
<td>Rank 8</td>
<td>0.1033</td>
<td>0.1871</td>
</tr>
<tr>
<td>Rank 9</td>
<td>-0.0128</td>
<td>0.2147</td>
</tr>
<tr>
<td>Rank 10</td>
<td>0.1257</td>
<td>0.2073</td>
</tr>
<tr>
<td>Rank 11</td>
<td>-0.1085</td>
<td>0.2340</td>
</tr>
<tr>
<td>Rank 12</td>
<td>0.3670</td>
<td>0.1860</td>
</tr>
<tr>
<td>Endorsement: Gold</td>
<td>-0.0022</td>
<td>0.1328</td>
</tr>
<tr>
<td>Endorsement: Silver</td>
<td>-0.1431</td>
<td>0.1493</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>-0.1236*</td>
<td>0.1044</td>
</tr>
<tr>
<td>Search Constant</td>
<td>-6.5377***</td>
<td>0.1357</td>
</tr>
</tbody>
</table>
Counterfactual simulations
Conclusions

• We propose a double index framework that subsumes several search models

• We propose a rank optimization algorithm to maximize “ex ante” consumer surplus

• We study a real world business platform and improve its ranking algorithm
“Ex post” consumer surplus
Sequential search with recall

- Following Weitzman (1979), each product delivers an ex-ante unknown payoff $u_j \sim F_j^U$.

- Payoff can be learned by paying a cost $c_j$

$$c_j = \int_{s_j}^{\infty} (u - s_j) dF_j^U(u)$$

- Let $S$ denotes the set for unsearched products and $S^c$ as those already searched.

- Define $s^* = \max\{j \in S\} s_j$ and $v^* = \max_{j \in S^c} u_j$.

  - If $s^* > v^*$, the consumer keeps on searching and searches the option with search score $s^*$ first.

  - If $s^* \leq v^*$, the consumer stops searching and purchases the option with utility $v^*$.
Simultaneous search

- Ex-ante identical goods with payoffs drawn from $F_U$
- $F_U$ supported on $[u, \bar{u}]$
- Searching $n$ goods entails a cost $c(n)$
- Consumers optimally choose how many goods to sample, $n^*$
  - $n \rightarrow \uparrow$ search cost
  - $n \rightarrow$ better $u$ draw in expectation
- This strategy is subsumed in the double index model by setting $s_j = \bar{u}$ for $n^*$ randomly chosen goods and $s_j = u$ for the rest
Satisficing

- First introduced by Simon (1955)
- Ex-ante identical goods
- Consumers search sequentially until a level of utility $s^*$ is reached
- This strategy is subsumed in the double index model by setting $s_j = s^*$ for all $j$