Assortment Optimization under Variants of the Nested Logit Model

James Davis
Cornell University ORIE

Guillermo Gallego
Columbia University IEOR

Huseyin Topaloglu
Cornell University ORIE

August 24, 2012
Assortment Optimization Problem

Input:
- $n$ items
- Revenue for each item: $r_i$
- Set dependent probability of purchase for each item:

$$P(i \text{ is purchased when } S \text{ offered}) = P_i(S)$$

Output:
- Set $S$ that maximizes expected revenue $\sum_{i \in S} r_i P_i(S)$
High level idea:

- Customer chooses a nest, or leaves
- Customer purchases an item, or leaves
Model Parameters

- $m$ nests
- $n$ items
Item $j$ in nest $i$ has preference weight $v_{ij}$
Model Parameters

- Outer no purchase (0 revenue) has weight $v_0$
- No purchase (0 revenue) for nest $i$ has weight $v_{i0}$
Model Parameters

- Dissimilarity parameter $\gamma_i$ for nest $i$
$P_i(S)$ in Nested Logit Model

Notation:
- Offer set $S = S_1 \cup \ldots \cup S_m$
- $V_i(S_i) = v_{i0} + \sum_{j \in S_i} v_{ij}$

Probabilities:
- $P($selecting nest $i$) = $\frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_k V_k(S_k)^{\gamma_k}} \propto V_i(S_i)^{\gamma_i}$
- $P($selecting item $j$ | nest $i$ selected) = $\frac{v_{ij}}{V_i(S_i)} \propto v_{ij}$
- $P($selecting item $j$) = $\frac{v_{ij} V_i(S_i)^{\gamma_i - 1}}{v_0 + \sum_k V_k(S_k)^{\gamma_k}}$
$P_i(S)$ in Nested Logit Model

$S = S_1 \cup S_2 \cup S_3$
Expected Revenues in Model

Expected revenue given nest $i$ selected:

$$R_i(S_i) = \sum_j r_{ij} \frac{v_{ij}}{v_0 + \sum_{k \in S_i} v_{ik}}$$

Expected revenue from $S = S_1 \cup \ldots \cup S_n$:

$$\sum_i R_i(S_i) \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_k V_k(S_k)^{\gamma_k}}$$
Full Problem Statement

Input:

- \( m \) nests
- \( n \) items in each nest
- Revenue for item \( j \) in nest \( i \): \( r_{ij} \)
  - Assume \( r_{i1} \geq \ldots \geq r_{in} \)
- Preference weights:
  - Item \( j \) in nest \( i \): \( v_{ij} \)
  - Outer no purchase: \( v_0 \)
  - No purchase for nest \( i \): \( v_{i0} \)
- Dissimilarity parameter for nest \( i \): \( \gamma_i \)

Output:

- Set \( S = S_1 \cup \ldots \cup S_m \) that maximizes expected revenue

\[
\sum_i R_i(S_i) \frac{V_i(S_i)\gamma_i}{v_0 + \sum_k V_k(S_k)\gamma_k}
\]
Motivation

- Generalization of MNL model, addresses independence of irrelevant alternatives (Ben-Akiva and Lerman 1997)
- Compatible with utility maximization (McFadden 1974, 1981, Borsch-Supan 1990)
$v_{i0}$ Values Change Behavior of Problem

$v_{i0} = 0$ for all $i$

- Customers can’t leave nests
- Problem is more tractable

$v_{i0} > 0$ for some $i$

- Customers can leave nests
- Problem is less tractable
\( \gamma_i \) Values Change Behavior of Problem

- \( \gamma_i \leq 1 \) for all \( i \)
  - Products in the same nest compete with each other
  - Problem is more tractable

- \( \gamma_i > 1 \) for some \( i \)
  - Products in the same nest synergize with each other
  - Problem is less tractable
NP-hardness Depends on $v_{i0}$ and $\gamma_i$

- NP-hardness reductions based on subset-sum
- Pseudo-poly-time algorithm exists for all variants

<table>
<thead>
<tr>
<th>$v_{i0}$</th>
<th>$\gamma_i \leq 1$</th>
<th>$\gamma_i &gt; 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{i0} = 0$</td>
<td>Poly-time solvable</td>
<td>NP-hard</td>
</tr>
<tr>
<td>$v_{i0} &gt; 0$</td>
<td>NP-hard</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>
Approximability Depends on $v_{i0}$ and $\gamma_i$

Notation:

- $\rho$ is ratio between largest and smallest $r_{ij}$ in any nest
- $\kappa$ is ratio between largest and smallest $v_{ij}$ in a nest
- $\bar{\gamma}$ is largest $\gamma_i$

<table>
<thead>
<tr>
<th>$v_{i0}$</th>
<th>$\gamma_i \leq 1$</th>
<th>$\gamma_i &gt; 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Poly-time solvable</td>
<td>$\max{\rho, 2\kappa}$-appx</td>
</tr>
<tr>
<td>$&gt; 0$</td>
<td>2-appx, FPTAS</td>
<td>$2\kappa$-appx, PTAS for fixed $\bar{\gamma}$</td>
</tr>
</tbody>
</table>
Poly-time solvable when $v_{i0} = 0, \gamma_i \leq 1$

$S_i$ is Revenue Ordered
If $j \in S_i$ all items with revenue $\geq r_j$ are in $S_i$

Theorem 1
If $S = S_1 \cup \ldots \cup S_n$ is the optimal assortment then $S_i$ is revenue ordered for all $i$ when $v_{i0} = 0$ and $\gamma_i \leq 1$ for all nests $i$.

Theorem 2
There is a poly-time algorithm to find an optimal assortment when $v_{i0} = 0$ and $\gamma_i \leq 1$ for all nests $i$.

Theorem 2 does not follow directly from Theorem 1!
Problem Can be Written as Program

\[
\begin{align*}
\min x \\
\quad x \geq \max_S \left\{ \sum_i R_i(S_i) \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_k V_k(S_k)^{\gamma_k}} \right\}
\end{align*}
\]
Constraint for $S$ can be Manipulated

\[ x \geq \sum_i R_i(S_i) \frac{V_i(S_i)^{\gamma_i}}{v_0 + \sum_k V_k(S_k)^{\gamma_k}} \]

\[ x(v_0 + \sum_k V_k(S_k)^{\gamma_k}) \geq \sum_i R_i(S_i) V_i(S_i)^{\gamma_i} \]

\[ xv_0 \geq \sum_i R_i(S_i) V_i(S_i)^{\gamma_i} - x \sum_k V_k(S_k)^{\gamma_k} \]

\[ xv_0 \geq \sum_i V_i(S_i)^{\gamma_i} (R_i(S_i) - x) \]
LP Decomposes by Nests

\[
\begin{align*}
\min x \\
xv_0 & \geq \sum_i V_i(S_i)\gamma_i(R_i(S_i) - x) \quad \forall S
\end{align*}
\]

\[
\begin{align*}
\min x \\
xv_0 & \geq \sum_i y_{S_i} \\
y_{S_i} & \geq V_i(S_i)\gamma_i(R_i(S_i) - x) \quad \forall i \forall S_i
\end{align*}
\]
Poly-time solvable when \( v_{i0} = 0, \gamma_i \leq 1 \)

Theorem 1
If \( S = S_1 \cup \ldots \cup S_n \) is the optimal assortment then \( S_i \) is revenue ordered for all \( i \) when \( v_{i0} = 0 \) and \( \gamma_i \leq 1 \) for all nests \( i \).

Theorem 2
There is a poly-time algorithm to find an optimal assortment when \( v_{i0} = 0 \) and \( \gamma_i \leq 1 \) for all nests \( i \).

Proof.

- There are at most \( n \) configurations for each nest
- There are at most \( nm \) constraints in the LP
Revenue Threshold Exists

\[ \gamma x + (1 - \gamma)R(S^*) \]

- \( S^* \) is optimal for nest, \( x \) is total optimal revenue
- Items with revenue above threshold are in optimal solution
Proof by Contradiction

• Suppose \( S^* \) is not revenue ordered
  • \( V(S^*_+)\gamma(R(S^*_+) - x) \geq V(S^*)\gamma(R(S^*) - x) \)
  • \( V(S^-)\gamma(R(S^-) - x) \geq V(S^*)\gamma(R(S^*) - x) \)

\[ \gamma x + (1 - \gamma)R(S^*) \]
Proof by Contradiction

- Suppose $S^*$ is not revenue ordered
- $V(S^*_+)\gamma (R(S^*_+) - x) \geq V(S^*)\gamma (R(S^*) - x)$
- $V(S^*_-)\gamma (R(S^*_-) - x) \geq V(S^*)\gamma (R(S^*) - x)$
Proof by Contradiction

- Suppose $S^*$ is not revenue ordered
- $V(S^+)\gamma(R(S^+) - x) \geq V(S^*)\gamma(R(S^*) - x)$
- $V(S_-)\gamma(R(S_-) - x) \geq V(S^*)\gamma(R(S^*) - x)$
\[ V(S_+^*) \gamma(R(S_+^*) - x) \geq V(S_+^*) \gamma(R(S_+^*) - x) \]

\[ V(S_+^*) \gamma(R(S_+^*) - x) = V(S_+^*) \gamma\left(\frac{\sum_{j \in S_+^*} v_j r_j}{V(S_+^*)} - x\right) \]

\[ = \sum_{j \in S_+^*} v_j r_j - V(S_+^*) x \]

\[ = \frac{\sum_{j \in S_+^*} v_j r_j}{V(S_+^*)^{1-\gamma}} \]

\[ \geq \sum_{j \in S_+^*} v_j r_j - V(S_+^*) x + v_3 (1 - \gamma)(R(S_+^*) - x) \]

\[ \geq \frac{\sum_{j \in S_+^*} v_j r_j}{V(S_+^*)^{1-\gamma}} \]

---

**Introduction**

**Results**

**LP Formulation**

**Revenue Order**

**Computations**
\[ V(S^*_+)^\gamma(R(S^*_+) - x) \geq V(S^*)^\gamma(R(S^*) - x) \]

\[
= \frac{\sum_{j \in S^*_+} v_j r_j - V(S^*_+) x + v_3 (1 - \gamma)(R(S^*) - x)}{V(S^*_+)^{1-\gamma}}
\]

\[
= \frac{V(S^*) \left( \frac{\sum_{j \in S^*_+} v_j r_j}{V(S^*)} - x \right) + v_3 (1 - \gamma)(R(S^*) - x)}{V(S^*_+)^{1-\gamma}}
\]

\[
= \frac{(V(S^*) + (1 - \gamma)v_3)(R(S^*) - x)}{V(S^*_+)^{1-\gamma}}
\]

\[
= \frac{(V(S^*) + (1 - \gamma)v_3)(R(S^*) - x)}{V(S^*_+)^{1-\gamma} + (1 - \gamma) V(S^*)^{1-\gamma} V(S^*_+) - V(S^*)}
\]
\[ V(S^*_+) \gamma(R(S^*_+) - x) \geq V(S^*) \gamma(R(S^*) - x) \]

\[
= \frac{(V(S^*) + (1 - \gamma)v_3)(R(S^*) - x)}{V(S^*)^{1-\gamma} + (1 - \gamma)V(S^*)^{-\gamma}(V(S^*_+) - V(S^*))}
\]

\[
= \frac{(V(S^*) + (1 - \gamma)v_3)(R(S^*) - x)}{V(S^*)^{1-\gamma} + (1 - \gamma)V(S^*)^{-\gamma}v_3}
\]

\[
= \frac{(V(S^*) + (1 - \gamma)v_3)(R(S^*) - x)}{V(S^*)^{-\gamma}(V(S^*) + (1 - \gamma)v_3)}
\]

\[
= V(S^*) \gamma(R(S^*) - x)
\]
Computational Results

- \( m = 5, n = 25 \)
- \( \epsilon \) controls \( v_{ij} \) and \( r_{ij} \) gaps (smaller \( \epsilon \) yields larger gaps)
- \( \gamma^L \) is smallest \( \gamma_i \), \( \gamma^U \) is largest \( \gamma_i \)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( ([0.5, 1.5], 0.6) )</td>
<td>0.01</td>
<td>0.34</td>
<td>0.01</td>
<td>0.20</td>
</tr>
<tr>
<td>( ([0.5, 1.5], 0.5) )</td>
<td>0.02</td>
<td>0.38</td>
<td>0.02</td>
<td>0.31</td>
</tr>
<tr>
<td>( ([0.5, 1.5], 0.4) )</td>
<td>0.03</td>
<td>0.66</td>
<td>0.03</td>
<td>0.48</td>
</tr>
<tr>
<td>( ([0.5, 1.5], 0.3) )</td>
<td>0.05</td>
<td>0.81</td>
<td>0.04</td>
<td>0.78</td>
</tr>
<tr>
<td>( ([1.0, 2.0], 0.6) )</td>
<td>0.03</td>
<td>0.52</td>
<td>0.03</td>
<td>0.51</td>
</tr>
<tr>
<td>( ([1.0, 2.0], 0.5) )</td>
<td>0.05</td>
<td>0.71</td>
<td>0.04</td>
<td>0.71</td>
</tr>
<tr>
<td>( ([1.0, 2.0], 0.4) )</td>
<td>0.08</td>
<td>1.43</td>
<td>0.07</td>
<td>0.92</td>
</tr>
<tr>
<td>( ([1.0, 2.0], 0.3) )</td>
<td>0.13</td>
<td>2.70</td>
<td>0.11</td>
<td>1.31</td>
</tr>
<tr>
<td>( ([1.5, 2.5], 0.6) )</td>
<td>0.06</td>
<td>1.56</td>
<td>0.05</td>
<td>0.79</td>
</tr>
<tr>
<td>( ([1.5, 2.5], 0.5) )</td>
<td>0.09</td>
<td>1.63</td>
<td>0.08</td>
<td>1.03</td>
</tr>
<tr>
<td>( ([1.5, 2.5], 0.4) )</td>
<td>0.14</td>
<td>2.13</td>
<td>0.12</td>
<td>2.13</td>
</tr>
<tr>
<td>( ([1.5, 3.0], 0.3) )</td>
<td>0.25</td>
<td>4.21</td>
<td>0.19</td>
<td>2.42</td>
</tr>
<tr>
<td>( ([2.0, 3.0], 0.6) )</td>
<td>0.09</td>
<td>2.00</td>
<td>0.08</td>
<td>1.24</td>
</tr>
<tr>
<td>( ([2.0, 3.0], 0.5) )</td>
<td>0.14</td>
<td>2.47</td>
<td>0.12</td>
<td>1.82</td>
</tr>
<tr>
<td>( ([2.0, 3.0], 0.4) )</td>
<td>0.22</td>
<td>3.70</td>
<td>0.18</td>
<td>2.20</td>
</tr>
<tr>
<td>( ([2.0, 3.0], 0.3) )</td>
<td>0.38</td>
<td>5.45</td>
<td>0.29</td>
<td>3.26</td>
</tr>
</tbody>
</table>
**Open Questions**

- FPTAS for general case
- Remove dependence on LP
- Extend to cross nested logit model
Open Questions

- FPTAS for general case
- Remove dependence on LP
- Extend to cross nested logit model

Thank You!