Optimal product positioning with consideration of negative utility effect on consumer choice rule

X.G. Luo, C.K. Kwong, J.F. Tang, Y.L. Tu

About product positioning

• The key decision of the levels of attributes of a new product
  – In the early stages of product development
  – Considering the competitors’ products

• Why is it difficult?
  – Unknown consumers’ purchase behavior

• Discrete choice model
  \[ u_{ij} = v_{ij} + \epsilon_{ij} \]
  (Net utility = observed + unobserved)
  – Indices:  \( i \) consumer; \( j \) product
  – General rule to estimated the expected sales
  – With the given distribution of \( \epsilon_{ij} \)
Contribution of this paper

• Providing a mathematical model\(^{(1)}\) and solution algorithms\(^{(2)}\)
  – (1) To determine the specification of a product and its price
  – (2) Bi-search algorithm based on Interval analysis (BS-IA) and Interval-analysis-embedded Tabu Search (IAE-TS)

• Extension of the *multinomial logit* rule
  – Probability (or expected sales) of a consumer \(i\) to purchase \(j\)
    under the multinomial logit rule
    \[
    P_{ij} = \frac{\exp(v_{ij})}{1 + \sum_j \exp(v_{ij})}
    \]
  – Integrating a piecewise logit function
    \[
    P_i = \frac{\Gamma(U_i, p)}{\Gamma(U_i, p) + \sum_j \Gamma(U_{ij}, p)} \quad \text{where}
    \]
    \[
    \Gamma(U_i, p) = \begin{cases} 
      \exp(\mu(U - p)), & \text{if } U - p \geq 0 \\
      0, & \text{otherwise}
    \end{cases}
    \]
Major differences from the existing research

• Idea
  – Consumer purchase probability is 0 if $p > U_j$

• Comparison between the three rules
  – Deterministic vs. MNL vs. Piecewise logit function (Fig. 1)
  – $U^*$ the maximal utility of all competitive products
Model

- **Key assumptions**
  - Consumers are segmented
  - Single product model introduction

- **Formulation**

\[
\max \Pi = \sum_{i=1}^{l} Q_i \left( \frac{\Gamma(U_i, p)}{\Gamma(U_i, p) + \sum_{j=1}^{N} \Gamma(U_{ij}, p_j)} \right) (p - C^{\text{var}}) - C^{\text{fix}}
\]

subject to

\[
\sum_{l=1}^{L_k} x_{kl} = 1, \quad k = 1, 2, ..., K
\]

\[
U_i = \sum_{k=1}^{K} \sum_{l=1}^{L_k} u_{kl} x_{kl}, \quad i = 1, 2, ..., l
\]

\[
C^{\text{var}} = \sum_{k=1}^{K} \sum_{l=1}^{L_k} c_{kl}^{\text{var}} x_{kl}, \quad i = 1, 2, ..., l
\]

\[
x_{kl} = 0 \text{ or } 1, \quad k = 1, 2, ..., K, \quad l = 1, 2, ..., L_k; \quad p > 0
\]

**Notation**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>Consumer segment</td>
</tr>
<tr>
<td>(k)</td>
<td>Attribute</td>
</tr>
<tr>
<td>(l)</td>
<td>Attribute level</td>
</tr>
<tr>
<td>(p)</td>
<td>Price of the product</td>
</tr>
<tr>
<td>(x_{kl})</td>
<td>1 if the product has (l) for (k)</td>
</tr>
<tr>
<td>(Q_i)</td>
<td># of consumers in (i)</td>
</tr>
<tr>
<td>(C^{\text{var}})</td>
<td>Variable cost</td>
</tr>
<tr>
<td>(C^{\text{fix}})</td>
<td>Fixed cost</td>
</tr>
<tr>
<td>(c_{kl}^{\text{var}})</td>
<td>Cost to use (l) for (k)</td>
</tr>
</tbody>
</table>

: net-profit

: attribute level assignment

: utility calculation

: fixed cost determination

: binary and positivity constraints
Model

- Key assumptions
  - Consumers are segmented
  - Single product model introduction

- Formulation

\[
\text{Max } \Pi = \sum_{i=1}^{l} \frac{\Gamma(U_i, p)}{\Gamma(U_i, p) + \sum_{j=1}^{N} \Gamma(U_{ij}, p_j)} (p - C_{\text{var}}) - C_{\text{fix}}
\]

subject to

\[
\sum_{l=1}^{L_k} x_{kl} = 1, \quad k = 1, 2, \ldots, K
\]

\[
U_i = \sum_{k=1}^{K} \sum_{l=1}^{L_k} u_{kl} x_{kl}, \quad i = 1, 2, \ldots, I
\]

\[
C_{\text{var}} = \sum_{k=1}^{K} \sum_{l=1}^{L_k} c_{kl} x_{kl}, \quad i = 1, 2, \ldots, I
\]

\[
x_{kl} = 0 \text{ or } 1, \quad k = 1, 2, \ldots, K, \quad l = 1, 2, \ldots L_k; \quad p > 0
\]

: net-profit

: attribute level assignment

: utility calculation

: fixed cost determination

: binary and positivity constraints

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>Consumer segment</td>
</tr>
<tr>
<td>(k)</td>
<td>Attribute</td>
</tr>
<tr>
<td>(l)</td>
<td>Attribute level</td>
</tr>
<tr>
<td>(p)</td>
<td>Price of the product</td>
</tr>
<tr>
<td>(x_{kl})</td>
<td>1 if the product has (l) for (k)</td>
</tr>
<tr>
<td>(Q_i)</td>
<td># of consumers in (i)</td>
</tr>
<tr>
<td>(C_{\text{var}})</td>
<td>Variable cost</td>
</tr>
<tr>
<td>(C_{\text{fix}})</td>
<td>Fixed cost</td>
</tr>
<tr>
<td>(c_{kl})</td>
<td>Cost to use (l) for (k)</td>
</tr>
</tbody>
</table>
Solving approach (Small-scale 1: single segment)

• By enumeration flow, D.V.s $x_{kl}$ are regarded as given.

• Then, the model is degraded as

$$\text{Max } \Pi(p) = Q_1 \frac{e^{k(u_1-p)}}{e^{u_1-p}} + U_1(p-C_{\text{var}}), \quad U_1 > p > 0$$

where the only D.V. is the continuous variable $p$.

• **Theorem 1.** Its local optimal solution is a global optimal solution.
  – Proof by contradiction

• Traditional nonlinear programming algorithms can be used; such as Newton method and Fibonacci algorithm.
Solving approach (Small-scale 2: multi-segment)

• The aggregate of a set of unimodal functions is not necessarily unimodal.
  – Depending on the characteristics of each consumer segment

• Proposing Bi-Search algorithm, based on Interval Analysis (BS-IA, provide a sketch of the algorithm)
  1. Split the domain of \( p \) as \([X(1), X(2)], [X(2), X(3)], [X(3), X(4)], ...\)
  2. If \( L(\Pi'([a, b])) U(\Pi'([a, b])) \leq 0 \) and \( U(\Pi([a, b])) \geq \Pi^* \),
     set \( m = (a + b)/2 \) and \( \Pi^* = \Pi(m) \).
  3. Otherwise, prune the interval.
  4. With no more interval, stop.
Solving approach (Large-scale)

- Interval-Analysis-Embedded Tabu Search (IAE-TS)
  - Heuristic approach can be efficient
  - **Interval analysis** for \( p \), **Tabu search** for \( x_{kl} \)
  - Tabu search improves a given solution, avoiding potentially bad movements.

- Sketch of the algorithm
  1. (1) Let \( x = [x_1, x_2, ..., x_K] \) be a given solution.
  2. (2) Identify its neighborhoods; varying each \( x_k \) by one unit: e.g.,
     \[
     N([3, 1, 5]) = \{[2, 1, 5][4, 1, 5][3, 2, 5][3, 1, 4][3, 1, 6]\}
     \]
  3. (3) Identify the best solution \( x' \) within \( N(x) \).
  4. (4) Remember the change to obtain \( x' \), add the reverse change into a list (called tabu list).
  5. (5) With enough iterations, stop the algorithm.
     (refer to the paper for more understandings; such as Adaptive control of tabu list size, jump mechanism, and aspiration level check.)
Case study 1; application to digital camera (overview)

• Given attribute and attribute levels as Table 1
  – Company estimated variable costs

• Market survey
  – Three consumer segments whose sizes are 61,000, 33,000, and 42,000, respectively.
  – Estimated part-worth for each consumer segment (Table 4)

• Optimal product profile:
  – Pixel: 10; Anti-shock: Y; Screen: 3.0; Battery: Lith; Face detection: N; Mode: 18; Weight: 140–160; Zoom: 9; product price at HK$593
Case study 2; large-scale problems (overview)

- Used a randomly generated data (simulation)
  - Part-worth utility: random number generation from the uniform distribution between [$3, $6].
  - Variable cost: (similar) between [$1, $3]
  - Demand: (similar) between [1 million, 2 million]
  - Fixed cost: $5 million
  - Ten competitive products with random attribute levels

- Reference-based comparison to alternative 5 selection rules
  - Maximal part-worth utility (PU)
  - Maximal PU minus part-worth cost (PC)
  - Maximal average PU in all market segments
  - Maximal average PU minus PC in all market segments
  - Minimal PC
Case study 2; large-scale problems (overview; cont.)
Conclusion

• Conjoint-analysis based optimization model for product positioning is established
  – With the negative utility effect on consumer choice rule

• Analyzed the proposed model
  – Proposing algorithms for small- and large-sized cases