Randomized algorithms for lexicographic inference

R. Kohli [1] – K. Boughanmi [1] – V. Kohli [2]

[1] Columbia Business School

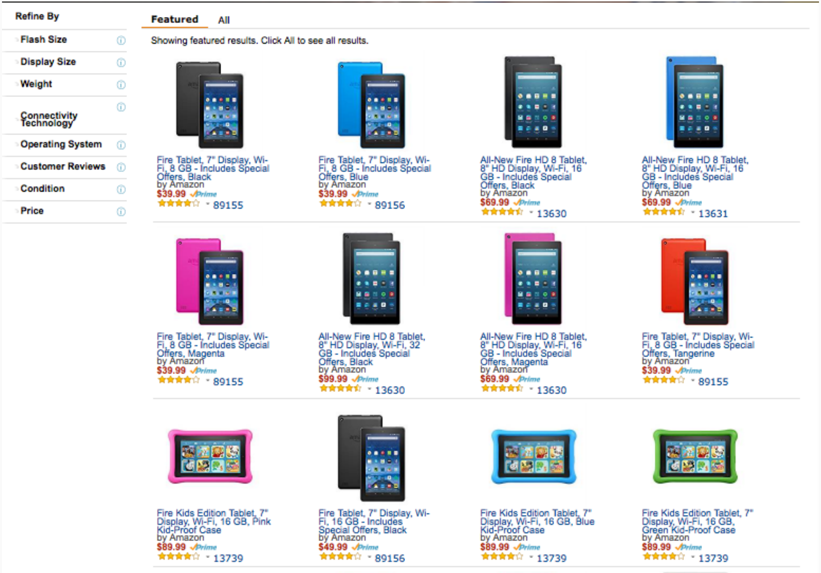
[2] Northwestern University

What is a lexicographic rule?

- 9.7-inch Retina display, wide color and true tone
- A9 third-generation chip with 64-bit architecture
- M9 motion coprocessor
- 1.2MP FaceTime HD camera
- 8MP iSight camera
- Touch ID
- Apple Pay

- 1.6 GHz exynos 7870 octa-core Processor
- Os: android 6.0 Marshmallow
- 2GB of RAM + 16GB
- 10.1" wuxga Display, 8Mp rear camera + 2Mp front camera
- Refer User manual in Technical Specification before use

Example of screening menu (lexicographic)



Lexicographic ranking of attributes levels

Rank	Attribute level
1	$169
2	$199
3	$299
4	iPad
5	128 GB hard drive
6	64 GB hard drive
7	4 GB RAM
8	$399
9	9" screen
10	Galaxy
...	

When is lexicographic screening useful?

- **Large choice sets**
  - Bröder (2000), Bröder and Schiffer (2003).
- **Break ties/avoid making tradeoffs**
  - Slovic (1975), Drolet and Luce (2004), Kohli and Jedidi (2007) and Yee et al. (2007).
- **Make judgments based on cues**
  - Hoffrage and Kleinbolting (1991), Bröder (2000), Gigerenzer, and Dieckmann, Dippold and Dietrich (2009).

Inference of lexicographic rules

- **Data**:
  - Paired-comparisons data (possibly inferred from rankings)
- **Objective**:
  - Find a lexicographic rule with maximal nb. of predictions consistent with the data (0-1 integer program)
- **"NP-Hard" problem**:
  - Computational difficulty increases with number of attributes/levels (n! possible sequences of n attribute levels)
- **Approximation algorithms**:
  - Greedy algorithm finds optimal solution only when there is a solution with perfect predictions

Schmitt and Martignon (2006), Kohli and Jedidi (2007), Yee, Dahan, Hauser and Orlin (2007)

Contribution

- Propose a Monte Carlo randomized algorithm for solving the lexicographic inference problem
  (e.g., Goemans and Williamson (1995) for maxcut problem)
- Maximizing expected value theoretically leads to the optimal solution
- Polynomial time maximum likelihood solution guarantees a lower bound on the performance
- The two solutions are related in the same way an arithmetic mean is related to a geometric mean
- Empirical implementation shows better results than previous methods

Presentation guideline

Discrete formulation

Randomized algorithm

Application: Choice of electronic tablets

Conclusion

Discrete formulation

Notations

Data

R: set of paired comparisons (data)

$r = (i, h) \in R$: alternative i preferred to h in the data

m: alternatives

t: attributes

n_l: number of levels of attribute l

$n = n_1 + \cdots + n_t$: total number of attribute levels

Notations (Cont.)

Decision variables

$x_{jk} = 1$ if level j is the k-th most important level, 0 otherwise

$$\sum_k x_{jk} = \sum_j x_{jk} = 1, \text{for all } 1 \leq j, k \leq n$$

Example: \$169 $\succeq$ iPad $\succeq$ Galaxy $\succeq$ 8" screen $\succeq$ 2GB RAM

Rank	iPad	\$169	Galaxy	2GB RAM	8" screen
1	0	1	0	0	0
2	1	0	0	0	0
3	0	0	1	0	0
4	0	0	0	0	1
5	0	0	0	1	0

0-1 integer programming formulation

$z_r = z_r(x) = 1$ if correct prediction for the pair $r \in R$, 0 otherwise

$$(P_1) \qquad \text{Maximize } z = \sum_{r \in R} z_r$$

$$\text{subject to}: \; 1 + \sum_{k=1}^{n} \sum_{j=1}^{n} \frac{1}{2^k} (a_{ij} - a_{hj}) x_{jk} \geq z_{ih}, \; \forall \, r = (i,h) \in R$$

$$\sum_{k=1}^{n} x_{jk} = \sum_{j=1}^{n} x_{jk} = 1, \quad \text{for all } 1 \leq j, k \leq n$$

$$z_{ih} \in \{0,1\}, \quad \text{for all } r = (i,h) \in R$$

$$x_{jk} \in \{0,1\}, \quad \text{for all } 1 \leq j, k \leq n$$

Computational complexity

Theorem

The lexicographic inference problem is NP-hard.

- **Generalization** of Schmitt and Martignon (2006)

- For m alternatives with n different levels:
  - $O(m^2 + n^2)$ 0-1 decision variables
  - $O(m^2 + n)$ linear constraints

- **Approximation algorithms are needed**
  - Greedy algorithm, local search

Randomized algorithm

Random utility formulation

For each level j assign a "utility" v_j

Suppose $v_1, \ldots, v_n$ are known

Consider the following randomized algorithm:

1. For each level j:
   (i) Obtain independent and random draw $\epsilon_j \sim E(0, 1)$
   (ii) Calculate $u_j = v_j + \epsilon_j$
2. Arrange the u_j in a decreasing sequence, $u_{j_1} > \cdots > u_{j_n}$
3. Consider the ordered sequence of levels $s = (j_1, \ldots, j_n)$ a candidate solution to lexicographic inference problem

Sequence probability

S: set of $n!$ possible sequences of attribute levels

Lemma

The randomized algorithm selects sequence $s \in S$ with probability

$$p(s) = p(u_{j_1} > \cdots > u_{j_n}) = \prod_{t=1}^{n-1} \frac{e^{v_{j_t}}}{e^{v_{j_t}} + \cdots + e^{v_{j_n}}}, \text{ for all } s \in S.$$

Expected correct prediction of the randomized solution

no. of correct predictions of one sequence s:

$$z(s) = \sum_{r \in R} z_r(s)$$

The expected value of the randomized solution for all possible sequence $s \in S$:

$$E = \sum_{s \in S} p(s) z(s).$$

Equivalence

$$(P_2') \qquad \underset{(v_j)\in\mathbb{R}^n}{\text{Maximize}} \ E = \sum_{s\in S} p(s)z(s).$$

Theorem

An optimal solution to problem P_2' is an optimal solution to the lexicographic inference problem P_1.

Solving (P_2') is impractical because it requires enumerating $n!$ sequences in S

Alternative expression for the expected value

	Levels				
	1	2	3	4	5
	iPad	8" screen	Galaxy	9" screen	4GB RAM
i	1	1	0	0	1
h	0	0	1	1	1

$$i \succcurlyeq h$$

Alternative expression for the expected value

	Levels				
	1	2	3	4	5
	iPad	8" screen	Galaxy	9" screen	4GB RAM
i	1	1	0	0	1
h	0	0	1	1	1

$$\underbrace{\underbrace{1 \quad 2}_{L_1^r} \quad \underbrace{3 \quad 4}_{L_2^r}}_{L_{12}^r}$$

$$
\begin{aligned}
p(r = (i, h)) &= \frac{e^{v_1}}{e^{v_1} + e^{v_2} + e^{v_3} + e^{v_4}} + \frac{e^{v_2}}{e^{v_1} + e^{v_2} + e^{v_3} + e^{v_4}} \\
&= \frac{e^{v_1} + e^{v_2}}{e^{v_1} + e^{v_2} + e^{v_3} + e^{v_4}} \\
&= \sum_{j \in L_1^r} \frac{e^{v_j}}{\sum_{\ell \in L_{12}^r} e^{v_\ell}}
\end{aligned}
$$

Better expression for E

An equivalent formulation:

$$(P_2) \qquad \operatorname*{Maximize}_{(v_j)\in\mathbb{R}^n} \; E = \sum_{r\in R}\sum_{j\in L_1^r} \frac{e^{v_j}}{\sum\limits_{\ell\in L_{12}^r} e^{v_\ell}}.$$

The new expression of E:

- enumerates only over $R \ll S$
- is a continuous unconstrained problem

But has a non convex objective function

Maximum likelihood solution

Obtain $\hat{v}_1, \ldots, \hat{v}_n$ by maximizing the likelihood function

$$\mathcal{L} = \prod_{r \in R} p(r).$$

Maximizing $\mathcal{L}$ is a convex optimization problem and can be solved in polynomial time.

Lower bound on the performance

N: the number of paired comparisons

$$E : \qquad a = \frac{E}{N} = \frac{1}{N} \sum_{r \in R} p(r)$$

denotes the arithmetic mean

$$\mathcal{L} : \qquad g = \mathcal{L}^{\frac{1}{n}} = \prod_{r \in R} p(r)^{1/N}$$

denotes the geometric mean

$$a \geq g \text{ implies } E \geq \hat{E} \geq N\hat{g} = N\hat{\mathcal{L}}^{\frac{1}{N}}$$

Lower bound on the performance

z^*: optimal solution to the lexicographic inference problem

Theorem

$$\frac{\hat{E}}{z^*} \geq \frac{N}{z^*}\left[\hat{g} + \min\left\{2\sigma^2, \frac{\kappa^2\sigma^2}{\kappa^2 - \sigma^2}\right\}\right] \geq \frac{1}{2}$$

where

$$\sigma^2 = \frac{1}{N}\sum_{i\in R}\left(\sqrt{\hat{p}(r)} - \mu\right)^2,\ \kappa^2 = \frac{1}{N}\sum_{r\in R}\left(\sqrt{\hat{p}_{\max}} - \sqrt{\hat{p}(r)}\right)^2,$$

$$\mu = \frac{1}{N}\sum_{r\in R}\sqrt{\hat{p}_i}, \quad \hat{p}_{\max} = \max_{i\in R}\hat{p}(r).$$

Randomized algorithm procedure

1. Maximize likelihood function and get: $\hat{v}_1, \ldots, \hat{v}_n$
2. Repeat for K iterations:
   - Draw $\epsilon_j \sim E(0,1)$ and compute $\hat{u}_j = \hat{v}_j + \epsilon_j$ for all j
   - Arrange the levels in a decreasing order given $u_{j_1} > \cdots > u_{j_n}$
   - Compute the number of correct predictions for sequence $s = (j_1, \ldots, j_n)$
3. Get the sequence $\hat{s}$ relative to $u = (u_{j_1} > \cdots > u_{j_n})$ with the highest correct predictions
4. Maximize E using u as a starting solution

Application: Choice of electronic tablets

Electronic tablets choice data

137 subjects

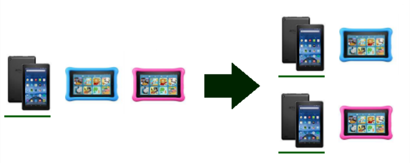
Each subject was shown fifteen choice sets

Each choice set had three alternatives

Task was to choose one alternative

Estimation

Paired comparisons were generated between the chosen alternative and the two other alternatives



$2 \times 15 \times 137 = 4{,}110$ paired comparisons

Aggregate analyses

Individual level analyses:

- Hierarchical Bayes (HMC)
- Maximum likelihood on individual data

Attributes and attributes levels

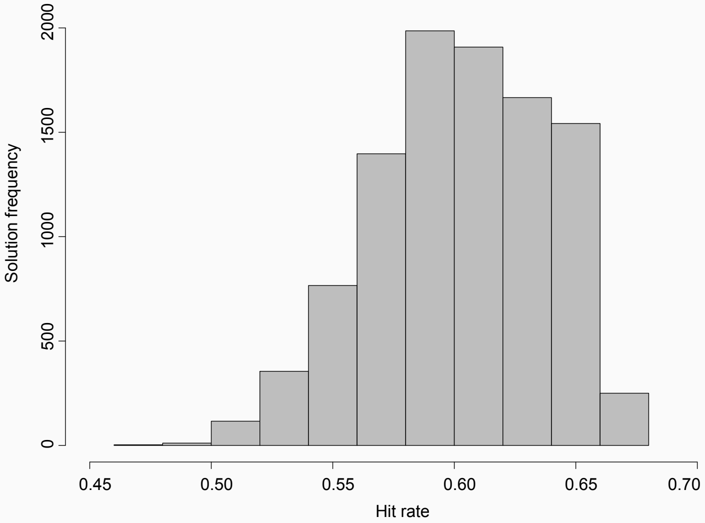
6 attributes and 24 levels

Categorical attributes		Ordinal attributes			
Brand	Screen size	Hard drive	RAM	Battery life	Price
iPad	7 inches	16GB	1GB	7 hours	\$169
Galaxy	8 inches	32GB	2GB	8 hours	\$199
Surface	9 inches	64GB	4GB	9 hours	\$299
Nexus	10 inches	128GB			\$399
Kindle					\$499

Lexicographic ranking of attribute levels (aggregate level)

Rank	Attribute level	Rank	Attribute level
1	$169	11	10" screen
2	$199	12	32 GB hard drive
3	$299	13	2 GB RAM
4	iPad	14	8" screen
5	128 GB hard drive	15	Surface
6	64 GB hard drive	16	Kindle
7	4 GB RAM	17	9-hour battery
8	$399	18	8-hour battery
9	9" screen	19	Nexus
10	Galaxy	20	7" screen

Randomized algorithm: Distribution of hit rates (1,000 iterations for aggregate level)



Hit rates (aggregate level)

Aggregate model	In sample	Out of sample
Logit model	0.604	0.602
Probit model	0.603	0.601
Greedy algorithm	0.657	0.655
Local search algorithm	0.669	0.668
Maximum likelihood solution	0.652	0.652
Randomized algorithm	0.671	0.652
Maximum expected value solution	0.677	0.676

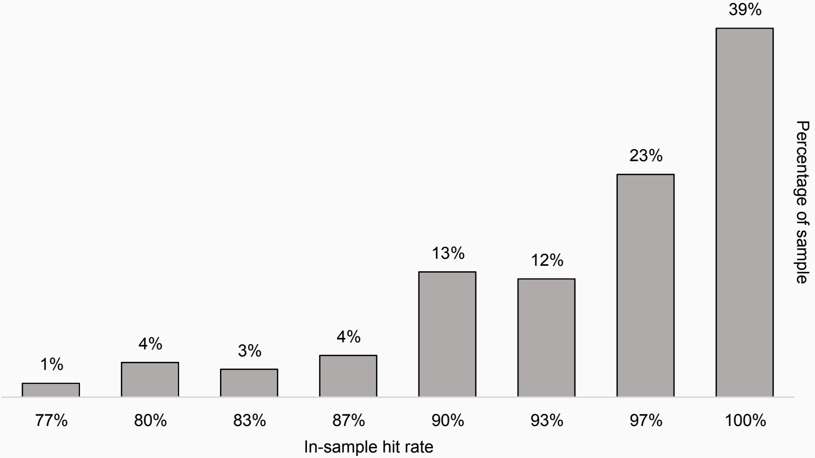
Computational time (aggregate level)

Model	CPU time
Logit model	0.18
Probit model	0.16
Greedy algorithm	0.03
Local search algorithm	13.87
Maximum likelihood solution	1.33
Randomized algorithm	4.14
Maximum expected value solution	10.58

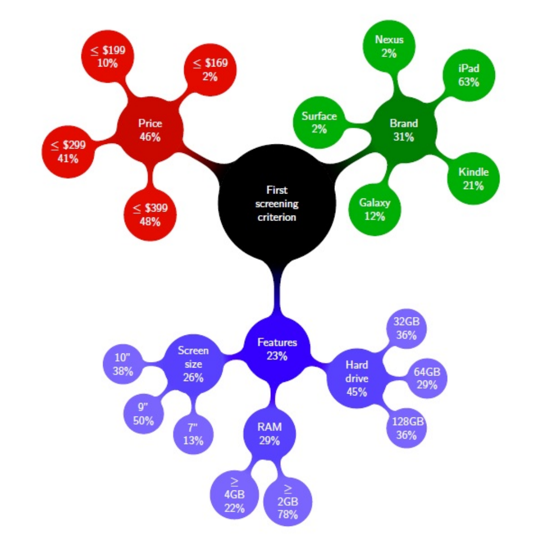
Hit rates (individual-level)

Model	In sample	Out of sample
Mixed logit model	0.844	0.764
Mixed probit mode	0.858	0.769
Greedy algorithm	0.873	0.784
Local search algorithm	0.915	0.768
Maximum likelihood solution	0.923	0.808
Randomized algorithm	0.947	0.818
Maximum expected value solution	0.949	0.816

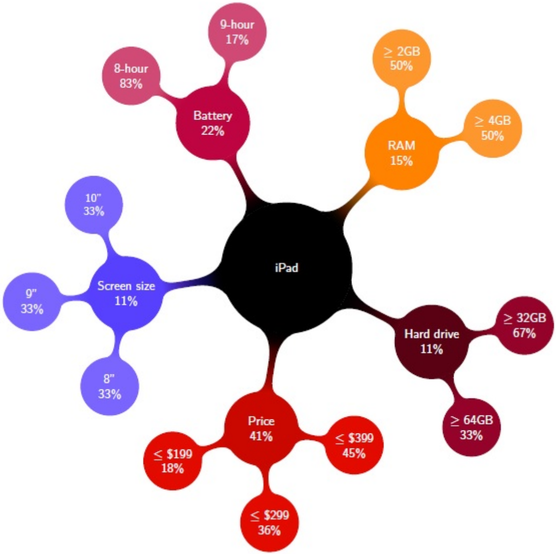
Randomized algorithm: Distribution of hit rates



First criterion used by different individuals



Second criterion used by individuals who use iPad as the first criterion



How the absence of iPad affects first-stage screening by individuals who use iPad at the first stage

Conclusion

Conclusion

- Proposed a Monte Carlo randomized algorithm for solving the lexicographic inference problem
- Maximizing expected value theoretically leads to the optimal solution
- Polynomial time maximum likelihood solution guarantees a lower bound on the performance
- The two solutions are related in the same way an arithmetic mean is related to a geometric mean
- Empirical implementation showed better results than previous methods

Thank you

0-1 integer programming formulation (cont.)

- $a = (a_{i1}, \ldots, a_{in})$: profile vector of alternative i
- $b_{ik} = \sum_{j=1}^{n} a_{ij} x_{jk}$ equal to a_{ij} if level j is the k-th most important attribute for alternative i
- $b_i = \sum_{k=1}^{n} \frac{b_{ik}}{2^k}$: n-digit binary number

A lexicographic ordering obtains a **non-reversal** for $r = (i, h)$ if it predicts correctly that i is preferred over h:

$$b_i - b_h = \sum_{k=1}^{n} \sum_{j=1}^{n} \frac{1}{2^k} (a_{ik} - a_{hj}) x_{kj} \geq 0$$

Aspect coding for ordinal attributes

	At most				
	$169	$199	$299	$399	$499
$169	**1**	**1**	**1**	**1**	**1**
$199	0	**1**	**1**	**1**	**1**
$299	0	0	**1**	**1**	**1**
$399	0	0	0	**1**	**1**
$499	0	0	0	0	**1**