Constructed, Augmented MaxDiff

Method and Case Study

Chris Chapman, Principal Researcher, Chromebooks
Eric Bahna, Product Manager, Android Auto

August 19, 2019
International Choice Modeling Conference, Kobe
Slides: https://bit.ly/2NfWPEA (2 N f [foxtrot] W P E A)

“I wish that I knew less about my customer’s priorities.”

“I wish that I knew less about my customer’s priorities.”

- No Product Manager Ever

Customer Input Becomes Feature Requests

Customer comments

Individual conversations

Usability studies

Surveys

Support forums

Conferences

Customer	Feature Request (FR)	Priority
CustomerA	FR1	P1
CustomerA	FR2	P1
CustomerA	FR4	P1
CustomerB	FR2	P0
CustomerC	FR3	P1
CustomerD	FR5	P1

Sparse, local data → global prioritization decisions

	FR1	FR2	FR3	FR4	FR5	FR6
CustomerA	P1	P1		P1		
CustomerB		P0				
CustomerC			P1			
CustomerD					P1	

PMs

Rank	**Feature**	**Priority**
1	FR2	P0
2	FR1	P0
3	FR4	P1
4	FR5	P1
5	FR3	P2
6	FR6	P2

Dense, global data → global prioritization decisions

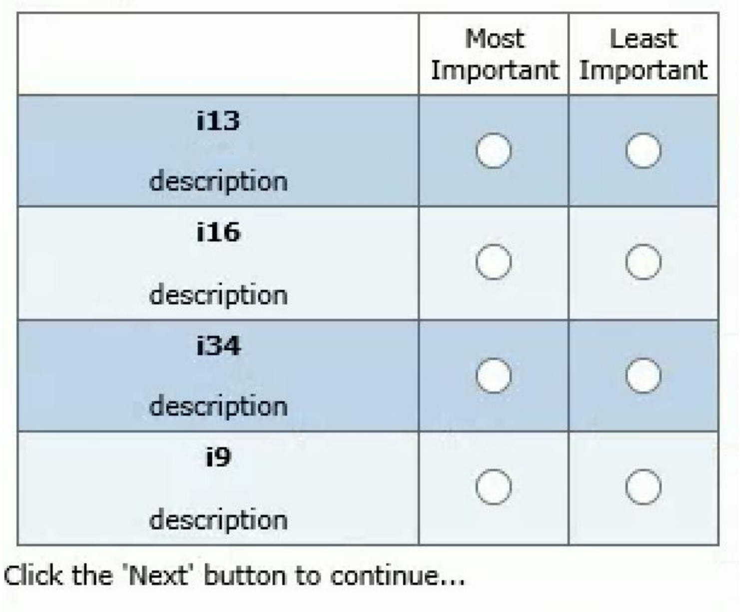
	FR1	FR2	FR3	FR4	FR5	FR6
CustomerA	P1	P1		P1		
CustomerB		P0				
CustomerC			P1			
CustomerD					P1	

	FR1	FR2	FR3	FR4	FR5	FR6
CustomerA	16	11	17	21	24	11
CustomerB	26	2	8	25	12	27
CustomerC	5	15	6	42	23	9
CustomerD	3	11	8	28	23	27

PMs

Rank	**Feature**	**Priority**
1	FR4	P0
2	FR2	P0
3	FR5	P1
4	FR6	P1
5	FR1	P2
6	FR3	P2

We often use MaxDiff surveys to prioritize users' feature requests



Rank	Feature	Priority
1	FR2	P0
2	FR1	P0
3	FR4	P1
4	FR5	P1
5	FR3	P2
6	FR6	P2

But: Some Problems with Standard MaxDiff

- Data Quality & Item relevance
  - Larger companies → more specialization
- Respondent experience
  - “Tedious” and “long”
- Inefficient use of respondent input
  - Wasting time on irrelevant items
  - More valuable to differentiate amongst “best” items

Some other MaxDiff Options

- **Adaptive MaxDiff** (Orme, 2006):
  Tournament-style progressive selection of items. More complex to program, less focused at beginning of survey. By itself, doesn't solve "I don't do that."

- **Express MaxDiff** (Wirth & Wolfrath, 2012):
  Selects subset of items to show each respondent. No insight at individual level on non-selected items. Addresses a different problem (long item list).

- **Sparse MaxDiff** (Wirth & Wolfrath, 2012):
  Uses all items from a long list per respondent, with few if any repetitions across choices. Low individual-level precision. Addresses long item lists.

- **Bandit MaxDiff** (Orme, 2018):
  Adaptively samples within respondent based on prior responses, sampling more often for higher preference. Achieves better discrimination among preferred items with potentially fewer tasks.

Constructed Augmented MaxDiff (CAMD)

CAMD Adds Two Questions Before MaxDiff

“Relevant?”

	I **have visibility** into this feature's importance	I **do not have visibility** into this feature's importance.
i24 description	○	○
i27 description	○	○
i8 description	○	○
i12 description	○	○
i21	○	○

⋮

Yes → Add to constructed list

“Important at all?”

	At least somewhat important	**Not important**
i9 description	○	○
i13 description	○	○
i4 description	○	○
i24 description	○	○
i29 description	○	○
	At least	

⋮

No → Use to augment data, save choice time

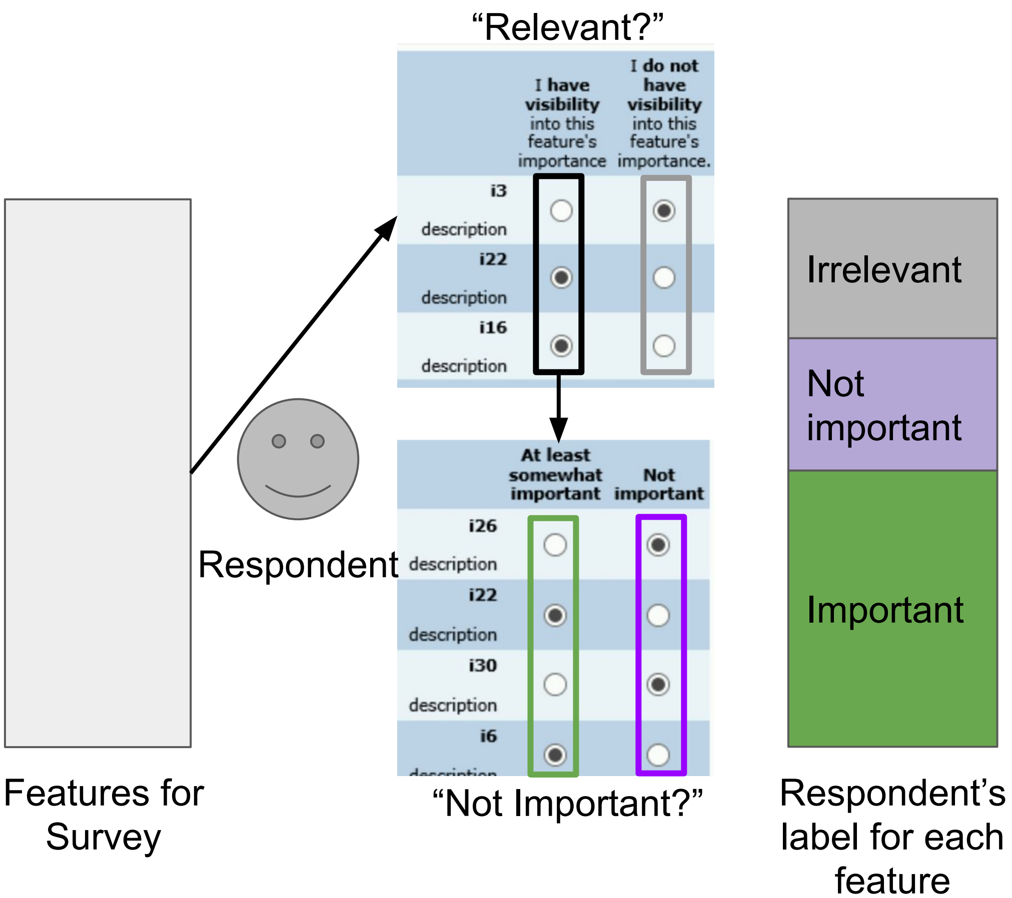
“Most & Least Important?”

	Most Important	Least Important
i13 description	○	○
i16 description	○	○
i34 description	○	○
i9 description	○	○

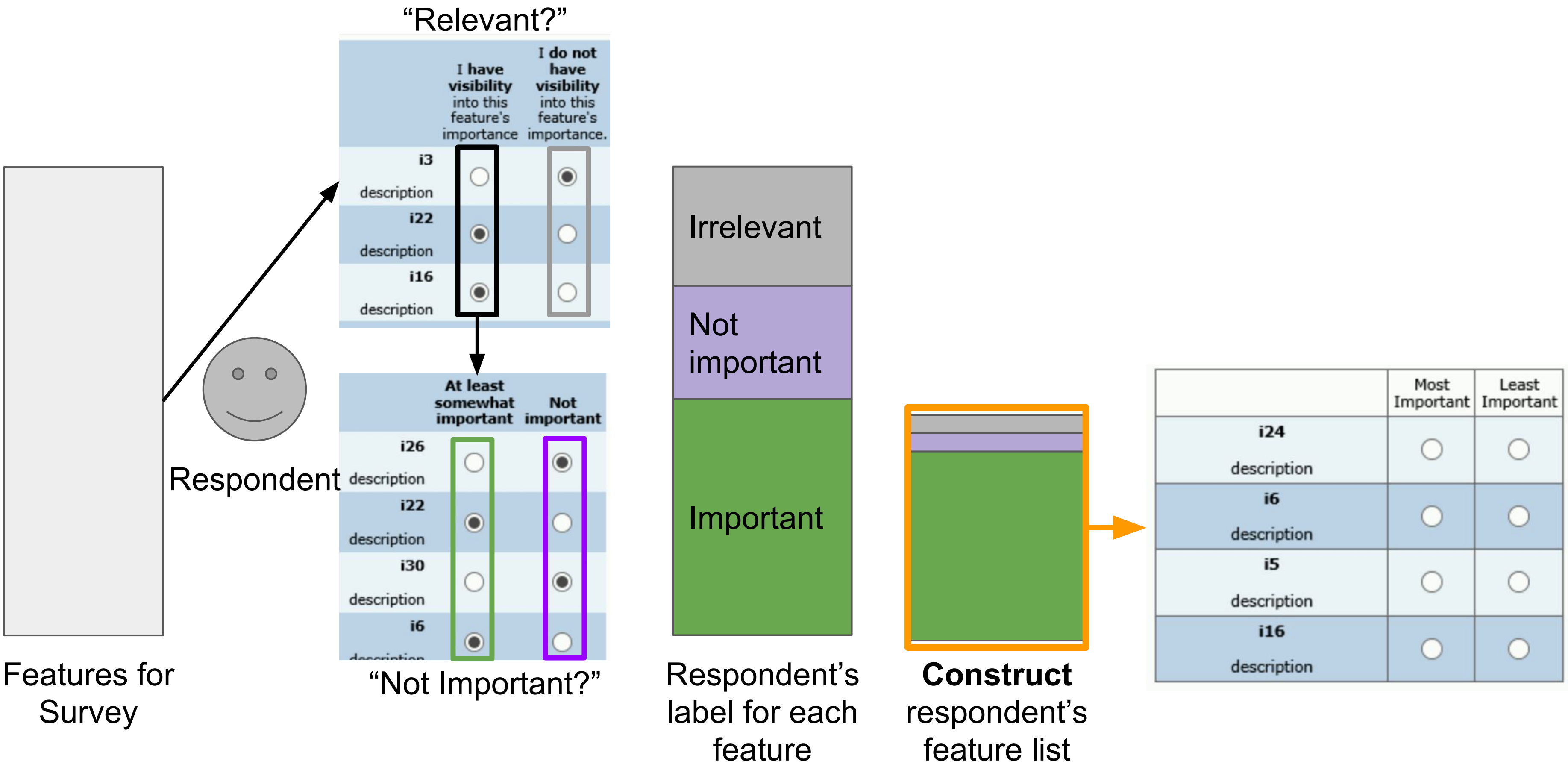
Click the 'Next' button to continue...

MaxDiff can use same task structure for all

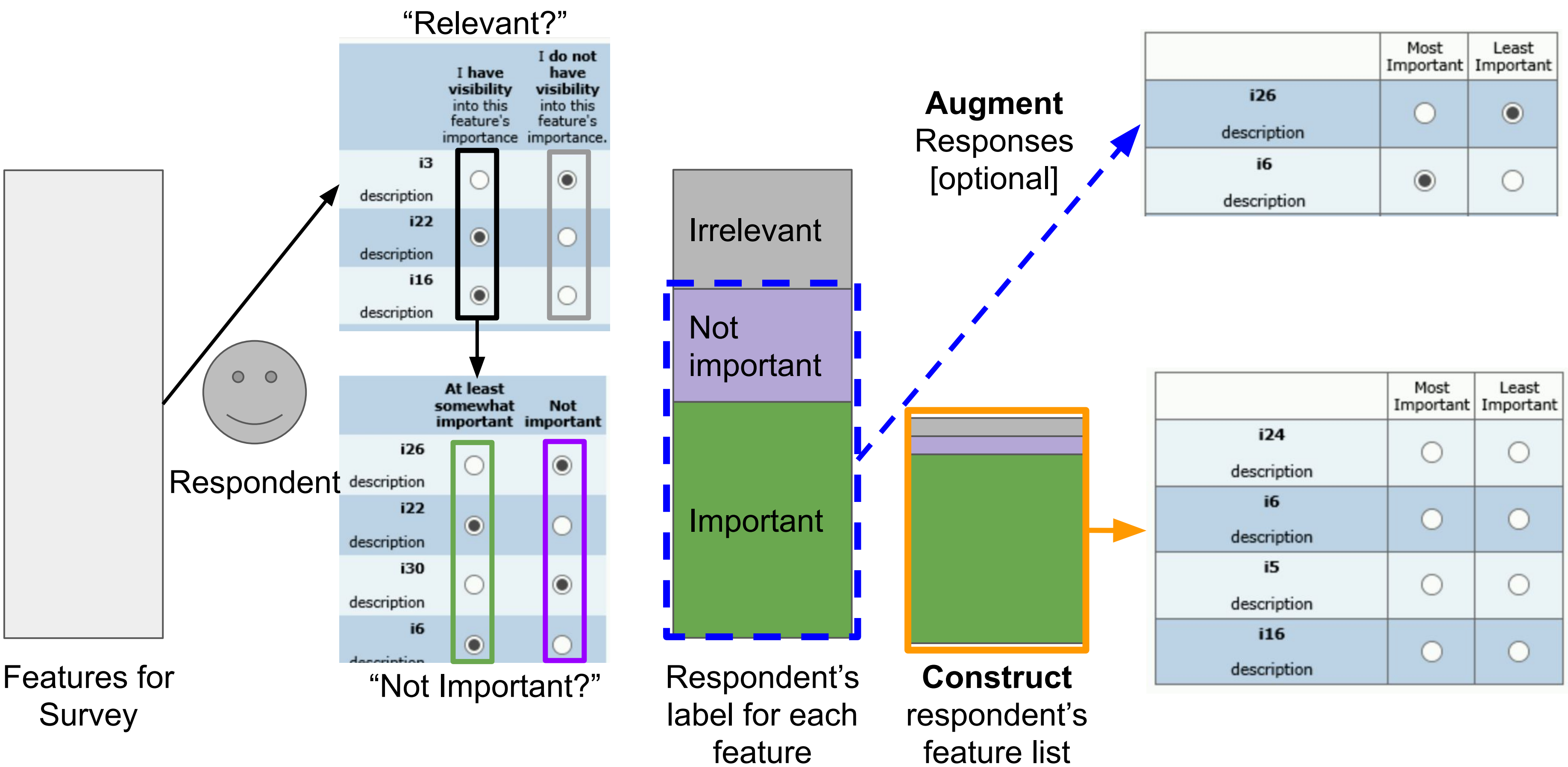
Constructed, Augmented MaxDiff



Constructed, Augmented MaxDiff



Constructed, Augmented MaxDiff



Threshold vs Grid Augmentation

For ***Relevant*** but ***Not Important*** items, we add implicit choice tasks:

A, B, C: Important
D, E, F: Not important

Full Grid augment
A > D
A > E
A > F
B > D
B > E
B > F
C > D
C > E
C > F … rapidly increases and augmented "tasks" may dwarf actual observations

Threshold vs Grid Augmentation

For ***Relevant*** but ***Not Important*** items, we add implicit choice tasks:

A, B, C: Important
D, E, F: Not important

Option:
Full Grid augment
A > D
A > E
A > F
B > D
B > E
B > F
C > D
C > E
C > F … rapidly increases and augmented "tasks" may dwarf actual observations

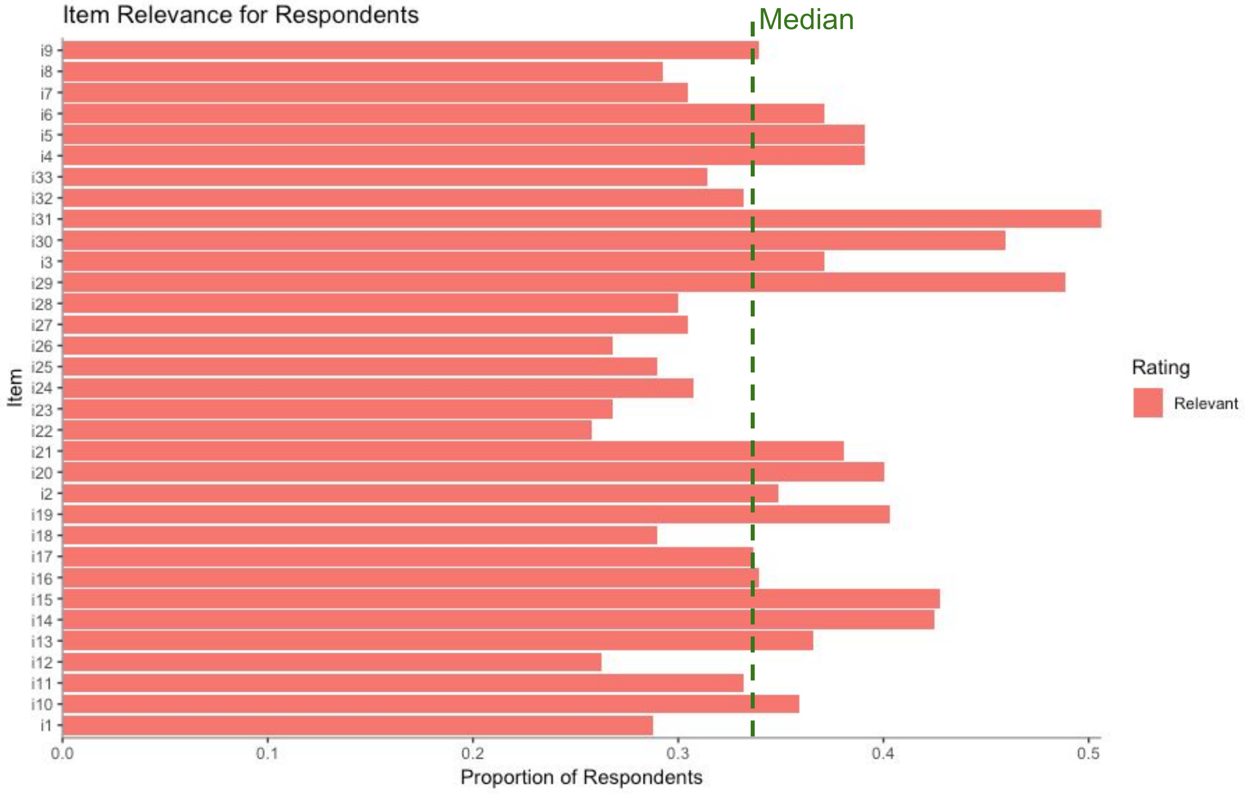
Recommended*:*
Threshold -- adds an implicit, latent "threshold" item
A > Threshold
B > Threshold
C > Threshold
Threshold > D
Threshold > E
Threshold > F … represents observed data with smaller addition of tasks

Results

Study with IT professionals

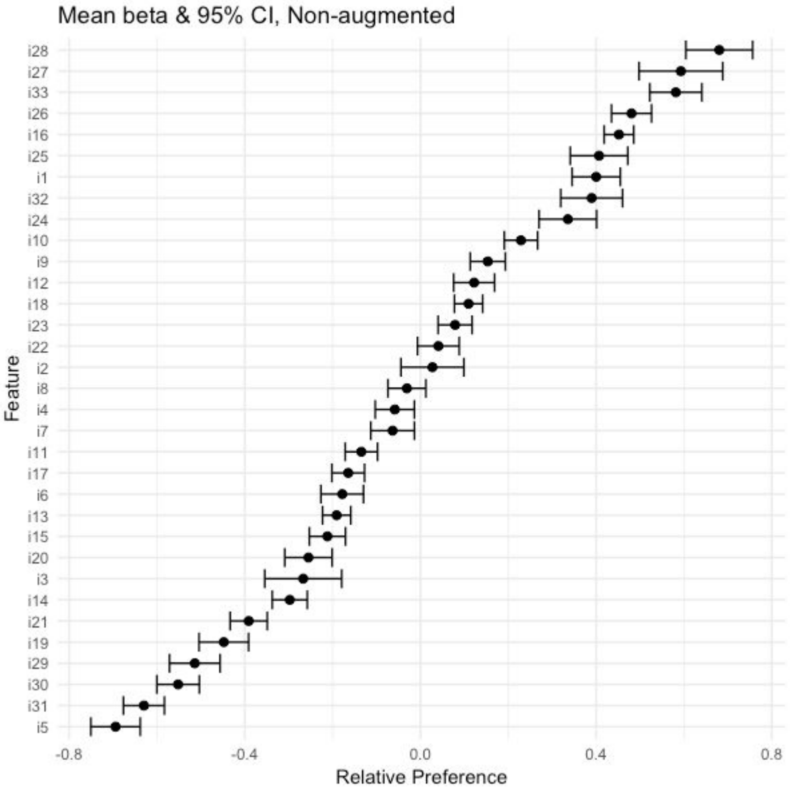
N=401 respondents, K=33 items

Results: 34% of Items Relevant to Median Respondent



Results: Before & After Augmentation

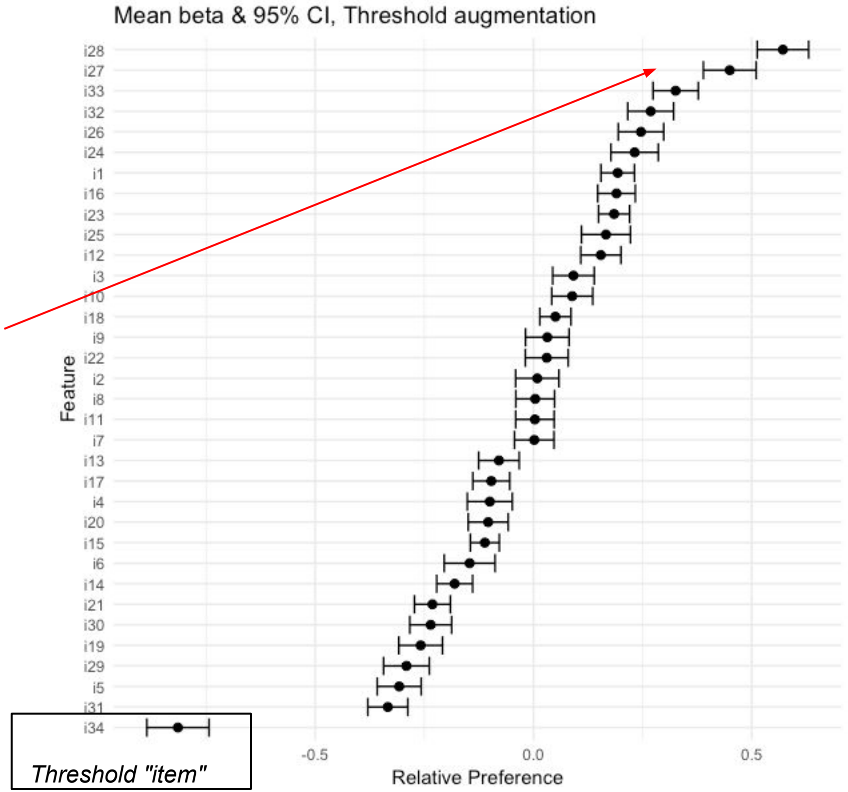
No Augmentation



Strong similarity in order

Threshold model has stronger discrimination at top

Threshold Augmentation

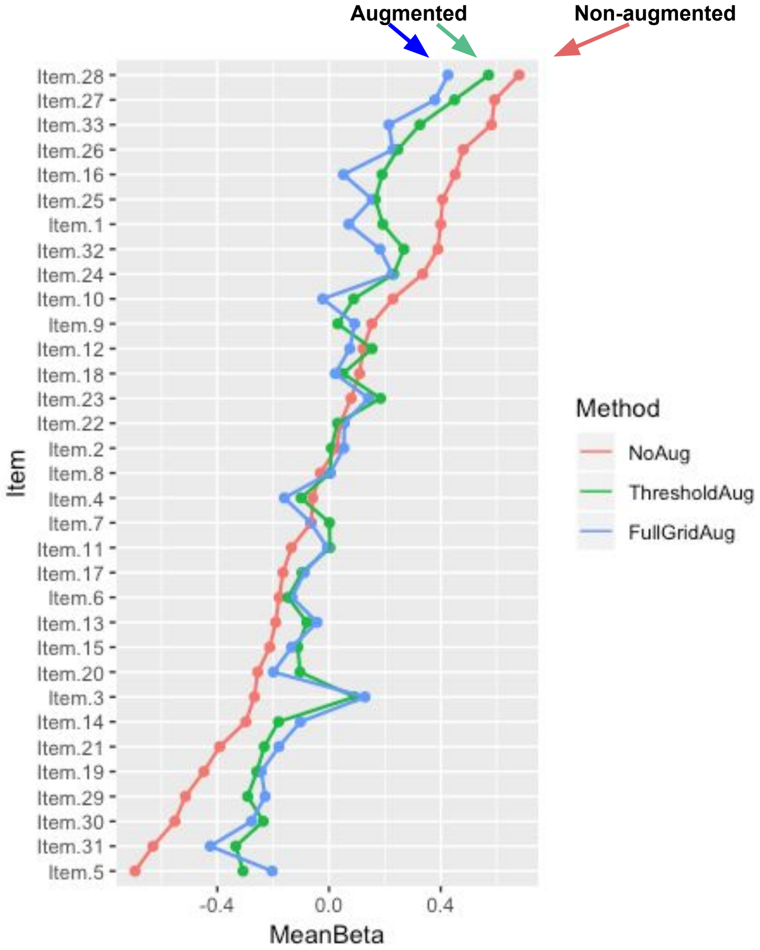


Results: Utilities Before and After Augmentation

- High overall agreement (r ~ 0.9+)
- Augmentation models are quite similar
- Augmentation may compress utilities
- Threshold augmentation is slightly more conservative vs. grid augmentation

Pearson's *r* values (between mean betas):

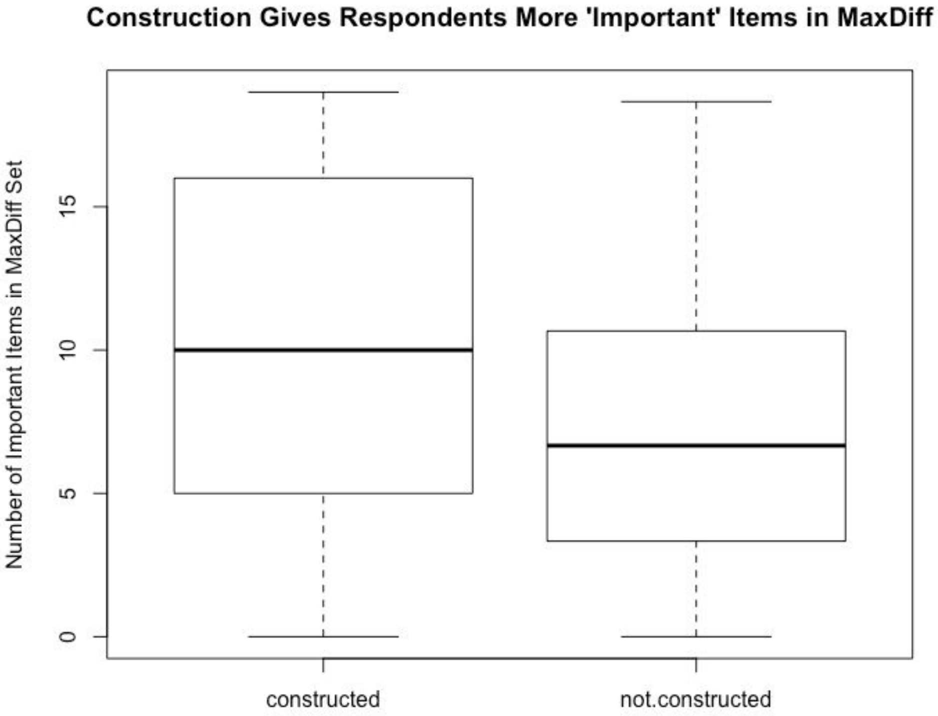
	NoAug	ThresholdAug	FullGridAug
NoAug	1.000		
ThresholdAug	**0.946**	1.000	
FullGridAug	**0.893**	**0.957**	1.000



Results: 50% More “Important” Items in MaxDiff

2nd study compared construction vs. non-constructed MaxDiff:

- Constructed MD study:
  - 30 items in survey
  - 20 items in MaxDiff exercise
- Without construction, we’d randomly select 20 of 30 items into MaxDiff exercise
- With construction, we emphasize “important” items



Results: Respondent and Executive Feedback

- Respondent feedback
  - “Format of this survey feels much **easier**”
  - “**Shorter** and **easier** to get through.”
  - “this time around it was a lot **quicker**.”
  - “Thanks so much for implementing the 'is this important to you' section! **Awesome** stuff!”

- Executive support
  - Funding for internal tool development
  - Advocacy across product areas
  - Support for teaching 10+ classes on MaxDiff, 100+ registrants

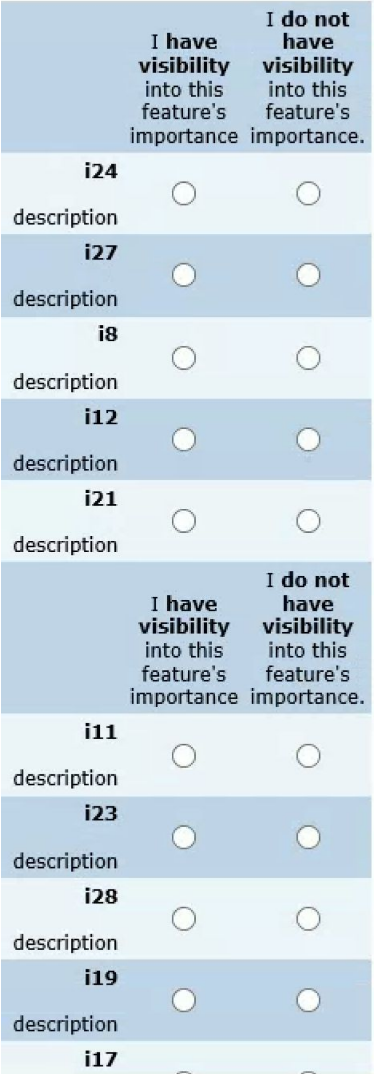
Discussion

Design Recommendations

- Initial rating for entire list of items, used to construct MaxDiff list
  Risk: Difficult to answer long list of "what's relevant"
  Solution: Break into chunks; ask a subset at a time; aggregate
  Could chunk within a page (as shown), or several pages.

- Construction of the MaxDiff list
  Risk: Items might be never selected ⇒ degenerate model
  Solution: Add 1-3 random items to the constructed list
  We used: 12 "relevant and important to me" +
  1 "not relevant to me" + 2 "not important"
  ⇒ MaxDiff design with 15 items on constructed list

- *Optional aspects:* Screening for "not relevant" items
  Including "not relevant" item(s) in tasks
  Augmentation



Open Topics (1)

- If respondents select the items to rate, what does "population" mean?
  Carefully consider what "best" and "worst" mean to you.
  Want: share of preference among **overall population**? ⇒ don't construct
  … *or*: share of preference among **relevant subset**? ⇒ construct

- Appropriate number of items -- if any -- to include randomly to ensure coverage
  We decided on 1 "not relevant" and 2 "not important", but that is a guess.
  Idea: Select tasks that omit those items, re-estimate, look at model stability.

- The best way to express the "*Relevant to you*?" and "*Important to you*?" ratings
  This needs careful pre-testing for appropriate wording of the task.

Open Topics (2)

- Construct separation, collinearity/endogeneity of relevance and importance
  Have seen evidence of high correlation in some cases; modest in others. Suspect dependence related to both domain and sample characteristics.

- Minimum # of relevant items needed in MD exercise?
  Model errors may be large if respondents differ greatly in # of relevant items. Suggest pre-testing to determine # of items to bring into the MD task.

- What if a P selects fewer than minimum # of relevant items?
  Two options: (1) *usually*: go ahead with MD and randomly selected tasks. (2) potentially: stack-rank exercise instead, create corresponding MD tasks (but: possibly overly coherent responses; endogeneity with item selection).

Demonstration of R Code

Referenced functions available at
https://github.com/cnchapman/choicetools

Features of the R Code

Data sources: Sawtooth Software (CHO file) ⇒ Common format
Qualtrics (CSV file) ⇒ Common format

Given the common data format:

⇒ **Estimation**: Aggregate logit (using `mlogit`)
Hierarchical Bayes (using `ChoiceModelR`)

⇒ **Augmentation**: Optionally augment data for "not important" implicit choices

⇒ **Plotting**: Plot routines for aggregate logit + upper- & lower-level HB

Example R Code: Complete Example

```
> md.define.saw <- list(                               # define the study, e.g.:
    md.item.k        = 33,                           # K items on list
    md.item.tasks    = 10,                           # num tasks (*more omitted)
...* )

> test.read <- read.md.cho(md.define.saw)              # convert CHO file
> md.define.saw$md.block <- test.read$md.block   # save the data

> test.aug <- md.augment(md.define.saw)                # augment the choices
> md.define.saw$md.block <- test.aug$md.block          # update data

> test.hb <- md.hb(md.define.saw, mcmc.iters=50000)    # HB estimation
> md.define.saw$md.hb.betas.zc <- test.hb$md.hb.betas.zc  # get ZC diffs


> plot.md.range(md.define.saw, item.disguise=TRUE)     # plot upper-level ests
> plot.md.indiv(md.define.saw, item.disguise=TRUE) +   # plot lower-level ests
    theme_minimal()                                    # plots = ggplot2 objects
```

Example R Code, Part 0: Define the Study

```
> md.define.saw <- list(                                        # define the study, e.g.:
    md.item.k        = 33,                                      # K items on list
    md.item.tasks    = 10,                                      # num of tasks
... )
```

Example R Code, Part 1: Data

```
> md.define.saw <- list(                                    # define the study, e.g.:
    md.item.k        = 33,                                # K items on list
    md.item.tasks    = 10,                                # num of tasks
... )

> test.read <- read.md.cho(md.define.saw)                   # convert CHO file
Reading CHO file: MaxDiffExport/MaxDiffExport.cho
Done. Read 407 total respondents.

> md.define.saw$md.block <- test.read$md.block              # save the data
```

Example R Code, Part 2: Augmentation

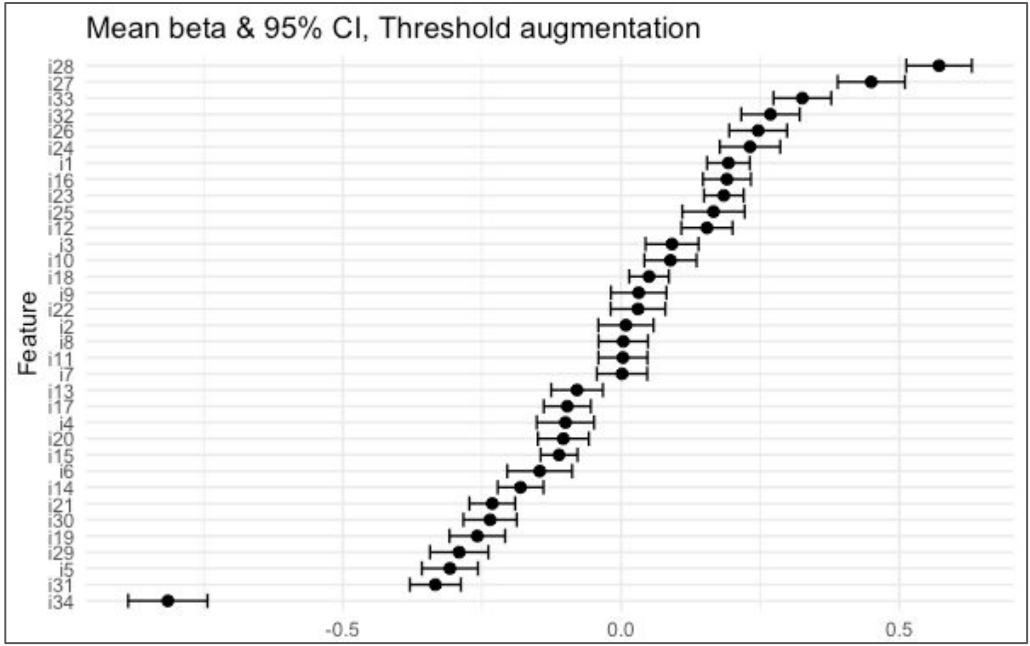
```
> md.define.saw$md.block <- test.read$md.block    # save the data
> test.aug <- md.augment(md.define.saw)           # augment the choices [optional]
Reading full data set to get augmentation variables.
Importants: 493 494 495 496 497 498 499 …
Unimportants: 592 593 594 595 596 597 …
Augmenting choices per 'adaptive' method.
Rows before adding: 40700

Augmenting adaptive data for respondent:
6  augmenting: 29 16 25 20 23 9 22 12 5 27 6 11 10 4 26 1 15 2 14 24 31 7 30
13 18 19 3 8 28 21 32 %*% 33 17 ...

Rows after augmenting data: 75640          # <== 1.8X data, 1x cost!

> md.define.saw$md.block <- test.aug$md.block     # update data with new choices
```

Example R Code, Part 3: HB

```
> md.define.saw$md.block <- test.aug$md.block          # update data with new choices

> test.hb <- md.hb(md.define.saw, mcmc.iters=50000)    # HB

MCMC Iteration Beginning…
Iteration  Acceptance    RLH      Pct. Cert.    Avg. Var.    RMS      Time to End
      100  0.339         0.483    0.162         0.26         0.31     83:47
      200  0.308         0.537    0.284         0.96         0.84     81:50 ...

> md.define.saw$md.hb.betas.zc <- test.hb$md.hb.betas.zc  # zero-centered diffs
```

Example R Code: Plots

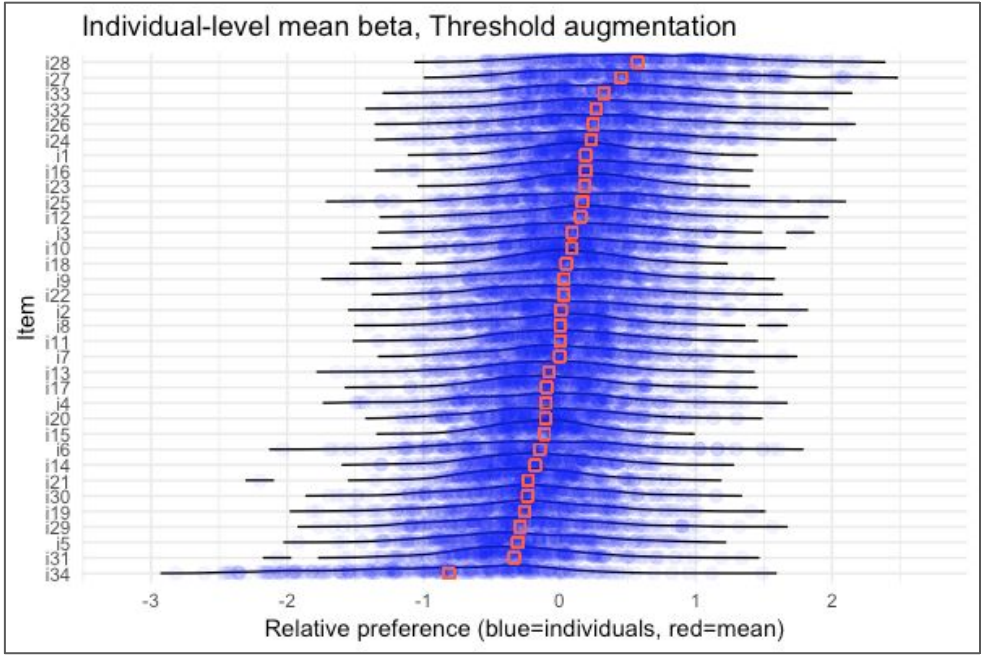
```
# upper-level range/CIs
> plot.md.range(md.define.saw,
                item.disguise=TRUE)
```



Example R Code: Plots

```
# upper-level
> plot.md.range(md.define.saw,
                item.disguise=TRUE)
```

```
# lower-level distribution
# note we can add ggplot2 functions
> plot.md.indiv(md.define.saw,
                item.disguise=TRUE) +
  theme_minimal()
```



Conclusions

- Higher quality data
  - Respondents were asked for MaxDiff input on more items that were relevant to them
- Better usage of data that respondents provided
  - We've observed 1.8 - 3.5x as many implicit choice tasks with augmented data
- Happier respondents
  - MaxDiff items were more relevant
  - We asked fewer MaxDiff questions because we could augment

- Use the code! Now an R package at GitHub as "cnchapman/choicetools"
  - For ***choice-based conjoint*** analysis, see UseR! 2019 presentation: http://bit.ly/2RO51fq

Thank you!

Constructed, Augmented MaxDiff: camd@google.com