Constructed, Augmented MaxDiff

EARL London
September 13, 2018

Chris Chapman Principal Researcher, Google
Eric Bahna Product Manager, Google

Overview

We often have lists of things we want customers to prioritize:

- Feature requests
- Key needs
- Product messaging
- Use cases and scenarios
- Generally, preferences amongst any set of things

Overview

We often have lists of things we want customers to prioritize:

Feature requests
Key needs
Product messaging
Use cases and scenarios
Generally, preferences amongst any set of things

We discuss how to do this systematically ...
... with shared R code, and modern Bayesian methods under the hood!

Problem: Sparse, local data vs. global prioritization

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

PMs

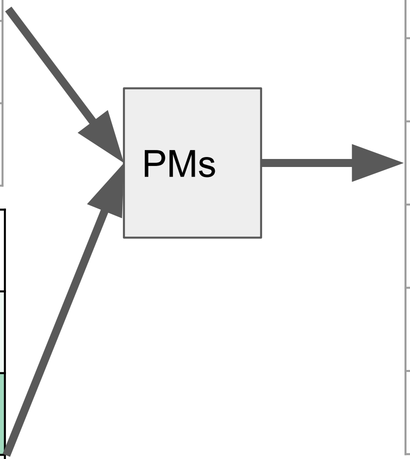
We want this ...

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



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

Rating scales don't work very well

Analysts often try to solve this problem with a rating scale:

How important is each feature?

	Not at all	*Slightly*	*Moderately*	*Very*	*Extremely*
Feature 1	☐	☐	☐	☐	☐
Feature 2	☐	☐	☐	☐	☐
Feature 3	☐	☐	☐	☐	☐
Feature 4	☐	☐	☐	☐	☐
Feature 5	☐	☐	☐	☐	☐

Rating scales don't work very well

Analysts often try to solve this problem with a rating scale:

How important is each feature?

	Not at all	*Slightly*	*Moderately*	*Very*	*Extremely*
Feature 1	☐	☐	☐	☐	☒
Feature 2	☐	☐	☐	☐	☒
Feature 3	☐	☐	☐	☐	☒
Feature 4	☐	☐	☐	☐	☒
Feature 5	☐	☐	☐	☐	☒

What's the problem? ⇒ No user cost: I can rate "everything is important!"
⇒ Not all "important" things are equally important

Common result: hard to interpret!

	Average Importance
Feature 1	4.6
Feature 2	4.3
Feature 3	4.4
Feature 4	4.8

Initial Solution: MaxDiff discrete choice survey

- Ask respondents to make **forced-choice tradeoffs** among features
- Repeat multiple times with **randomized** sets.
- Estimate a **mixed effects model** for overall and per-respondent preference

Considering just these 4 features, which one is
most important for you? Which one is **least important**?

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

Click the 'Next' button to continue...

⇒ *London EARL 2017 talk re discrete choice:*
https://goo.gl/73zasi

Concerns with Initial MaxDiff

Data Quality & Item relevance:

Enterprise respondents are often specialized; can't prioritize all items.

Respondent survey experience:

Length of survey is proportional to number of items. Shorter is better!

Concerns with Initial MaxDiff

Data Quality & Item relevance:
Enterprise respondents are often specialized; can't prioritize all items.

Respondent survey experience:
Length of survey is proportional to number of items. Shorter is better!

Solution:
Construct the MaxDiff list per respondent for what interests them.
Optionally ***augment*** the data file with inferred preferences.

⇒ Shorter surveys, better targeted, better differentiation of high priority items
⇒ "**Constructed, Augmented MaxDiff**" (CAMD).

[We admit it, not so catchy.]

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, saving time

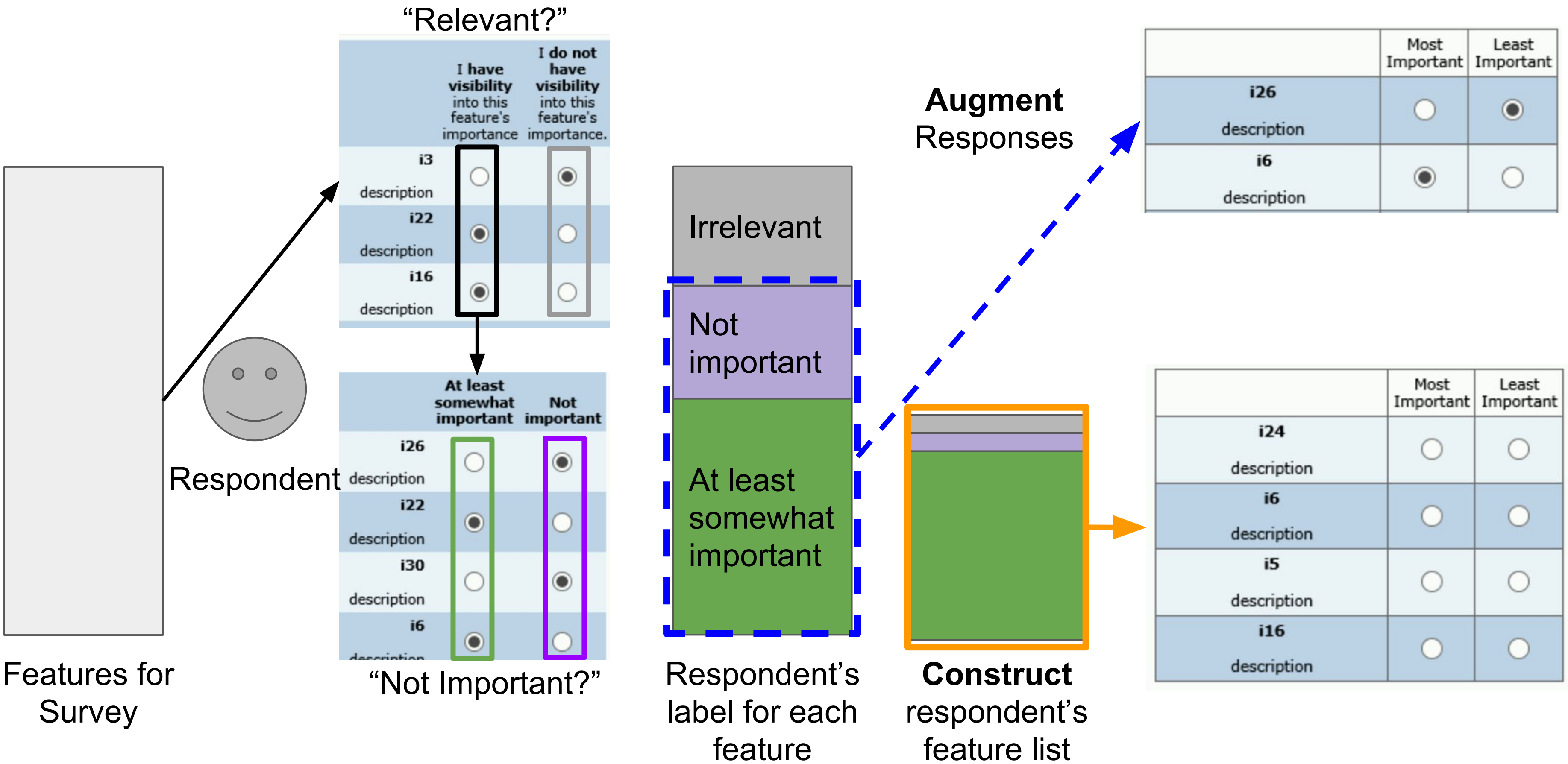
"Most & Least Important?"

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

Click the 'Next' button to continue...

MaxDiff uses the constructed list of items

CAMD Flow

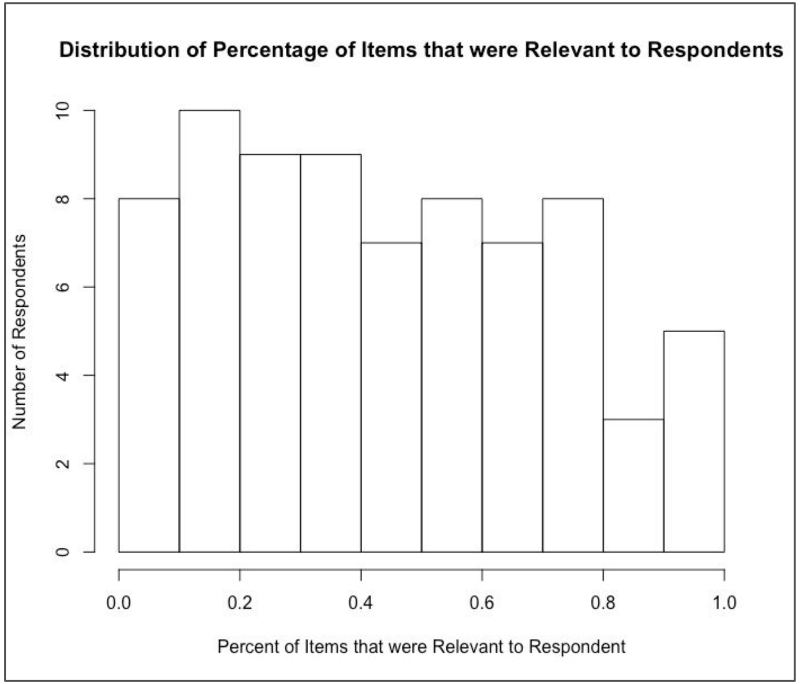


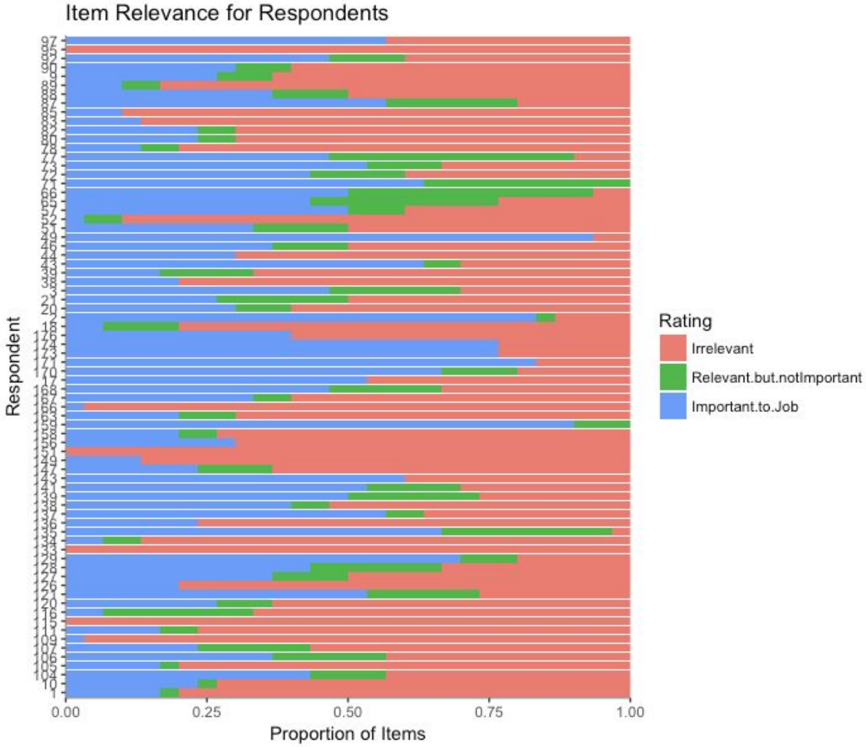
“Relevant?”
I have visibility into this feature's importance
I do not have visibility into this feature's importance.
i3
description
i22
description
i16
description
At least somewhat important
Not important
i26
description
i22
description
i30
description
i6
description
“Not Important?”
Respondent
Features for Survey
Irrelevant
Not important
At least somewhat important
Respondent’s label for each feature
Construct respondent’s feature list
Augment Responses
Most Important
Least Important
i26
description
i6
description
Most Important
Least Important
i24
description
i6
description
i5
description
i16
description

Results: Enterprise Feature Study

(items disguised)

Results: 55% of Items Irrelevant to Median Respondent

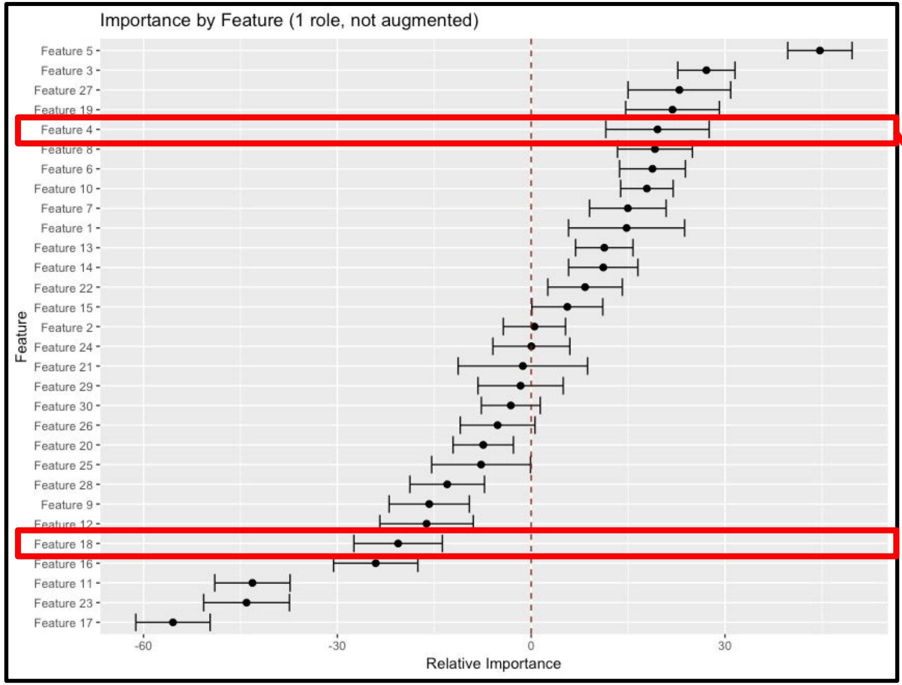




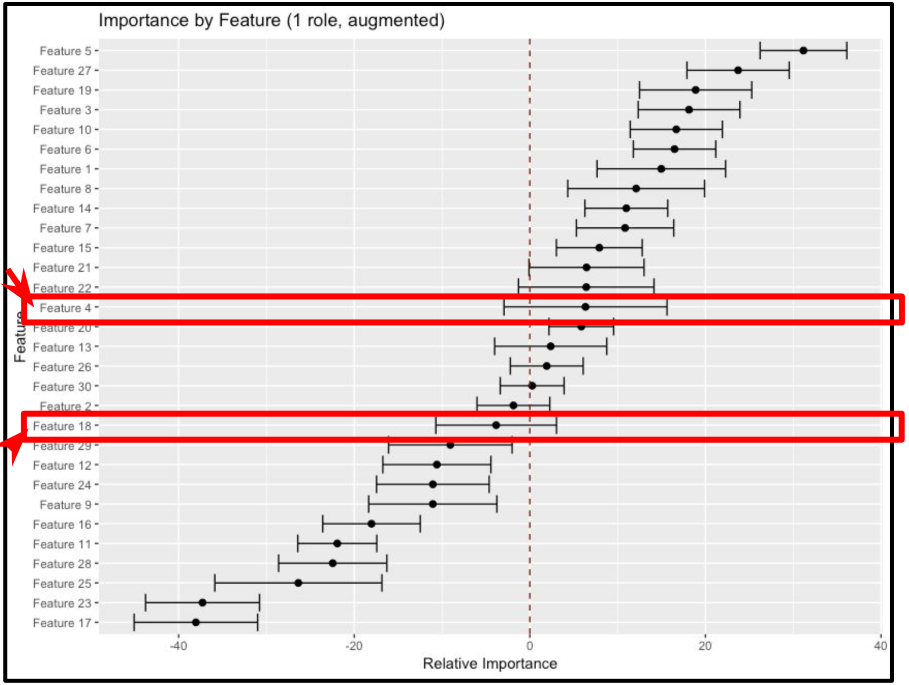
⇒ Huge time cost & dilution of data with noise if we ask about irrelevant items

Results: Before & After Augmentation

Before Augmentation



After Augmentation

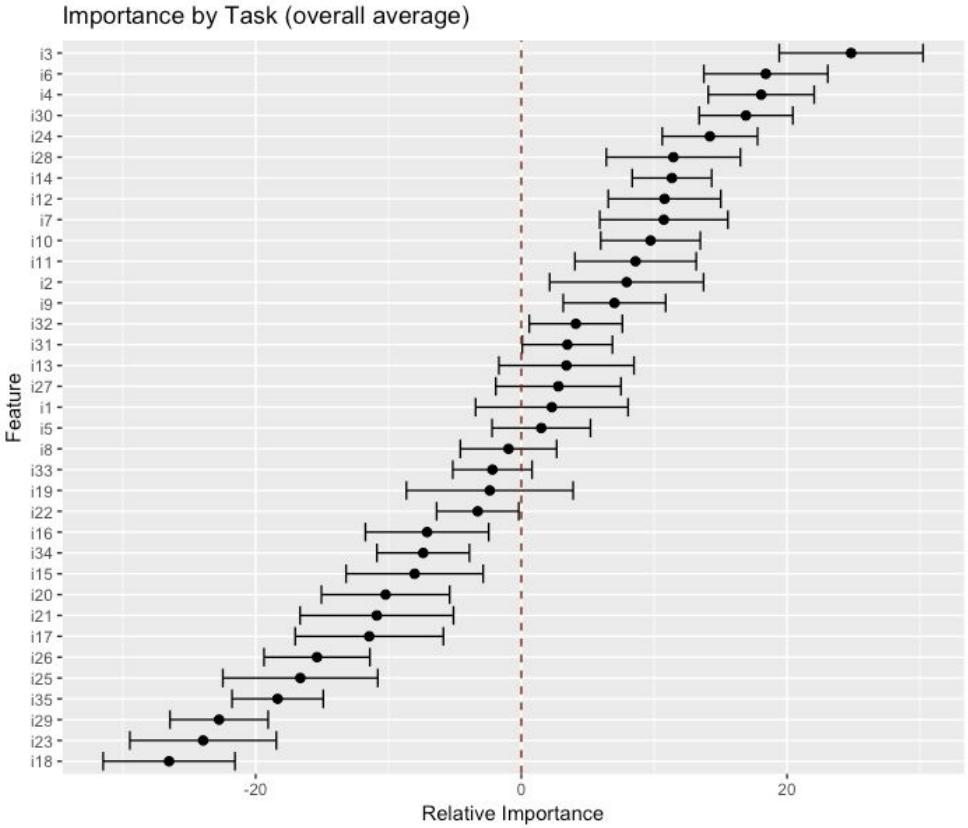


⇒ Modest changes; a few items change a lot, most don't. Good to use all the data!

Results: Changes in Business Priorities

Consider feature "i6" ...

Among 35 features, it was **#35 in engineering cost** to implement



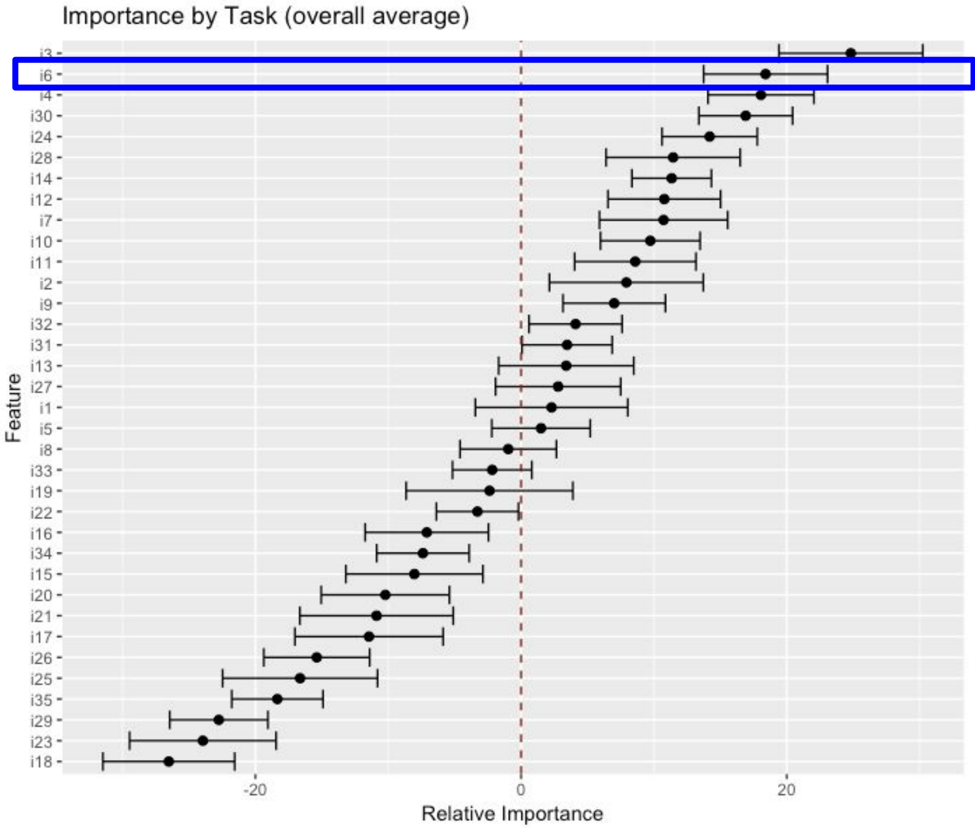
Results: Changes in Business Priorities

Consider feature "i6" …

Among 35 features, it was #35 in engineering cost to implement

… and now we learn that it is **#2 in overall customer priority**.

⇒ Much better coverage of customers' priorities, for a given amount of engineering resources



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 Googlers

- Surprise: many colleagues interested for internal use cases

R Code

Referenced functions available at [goo.gl/oK78kw](goo.gl/oK78kw)

Features of the R Code

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

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)                     # Sawtooth Software survey data
> md.define.saw$md.block <- test.read$md.block   # keep that in our study object

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

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

> plot.md.range(md.define.saw, item.disguise=TRUE)            # plot group-level estimates
> plot.md.indiv(md.define.saw, item.disguise=TRUE) +          # plot individual estimates
    theme_minimal()                                           # note plots use ggplot
```

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 Sawtooth 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

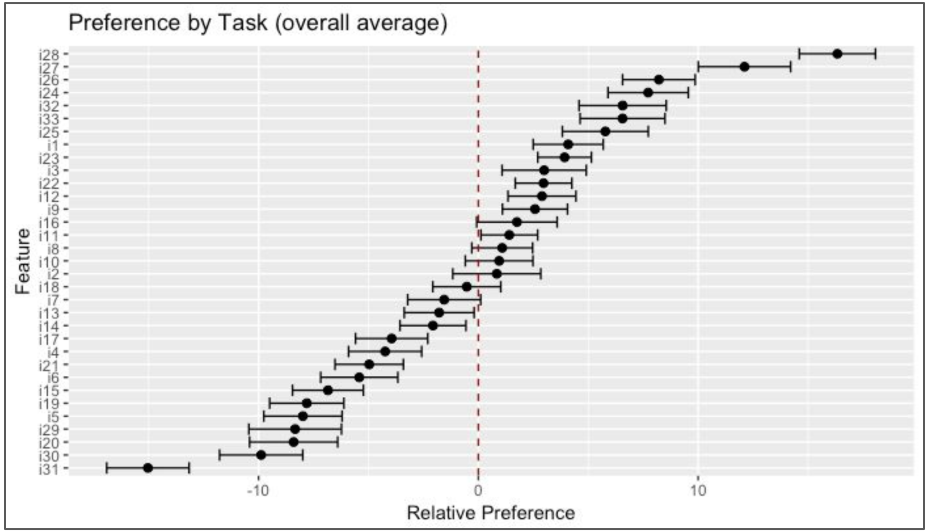
```
> md.define.saw$md.block <- test.read$md.block    # save the data
> test.aug <- md.augment(md.define.saw)           # augment the choices
Reading full data set to get augmentation variables.
Importants: 493 494 495 496 497 498 499 …
Unimportants: 592 593 594 595 596 597 …
Augmenting choices per 'adaptive' 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: 148660                # <== 3X data, 1x cost!

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

Example R Code, Part 3: HB

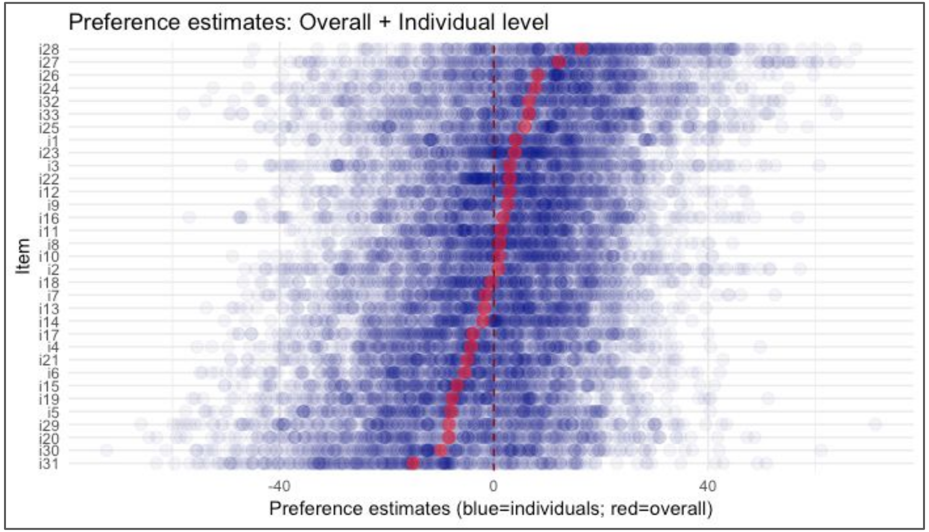
```
> md.define.saw$md.block <- test.aug$md.block         # update data with new choices

> 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
> plot.md.range(md.define.saw,
                item.disguise=TRUE)
```




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



Conclusions

- Higher quality data
  - Respondents are asked for input on more items that are relevant to them
- More data
  - We observed 2.0 - 3.5x as many implicit choice tasks with augmented data
- Happier respondents
  - MaxDiff items were more relevant to users
  - We asked fewer MaxDiff questions because we could augment the data

- Use the code! goo.gl/oK78kw

Thank you!

Constructed, Augmented MaxDiff: camd@google.com

Appendix: Additional findings

Some other MaxDiff Options

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

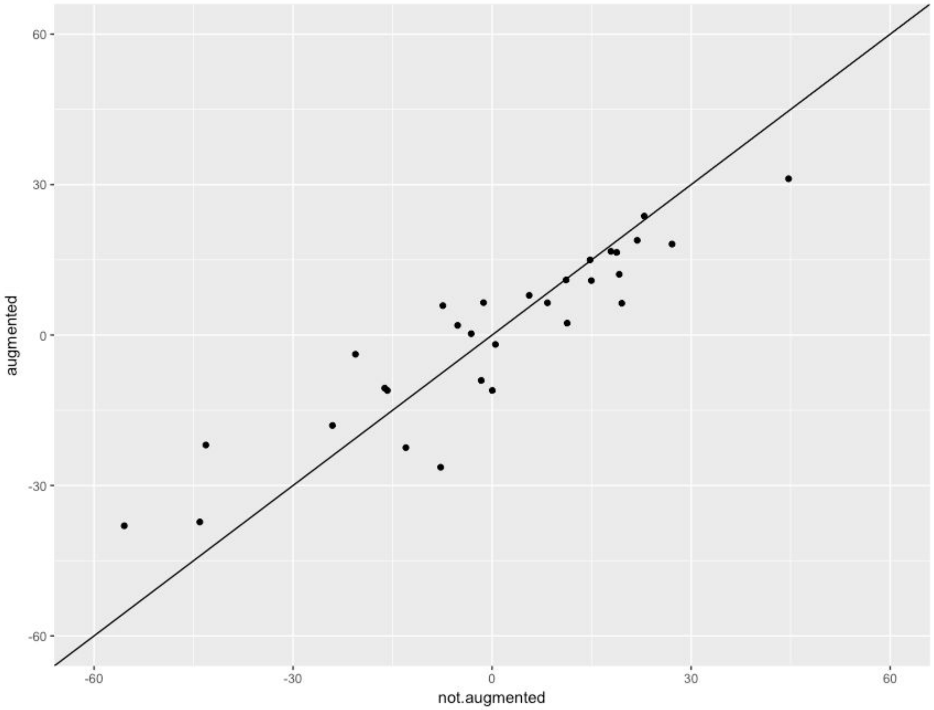
- Express MaxDiff (Wirth & Wolfrath, 2012):
  Selects subset of items to show each respondent. No insight at individual level on non-selected items. Addresses a different problem (long item list).

- 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 (Sawtooth Software, 2018):
  Focuses increasing attention on most-preferred items, based on previous choices. Addresses survey length concerns.

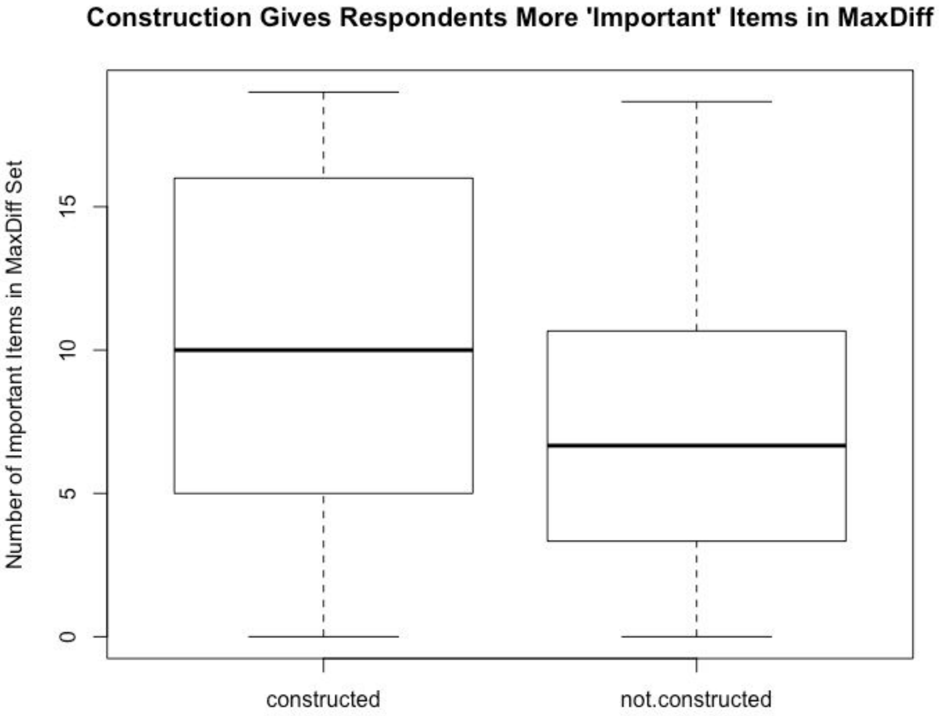
Results: Utilities Before and After Augmentation

- Modest adjustments to utilities
- Pearson's r = 0.90 between augmented and non-augmented utilities in one study
- Interesting that utilities became more compressed



Results: 50% More “Important” Items in 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



Appendix:
Additional Discussion and Design Recs

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

	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 description	○	○

	I have visibility into this feature's importance	**I do not have visibility** into this feature's importance.
i11 description	○	○
i23 description	○	○
i28 description	○	○
i19 description	○	○
i17		

Open Topics

- 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.

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