Prototyping Preference Inference Using Stan

John Mount
Win Vector LLC
https://www.win-vector.com/
September 10, 2024

Win Vector LLC

- Win Vector LLC is a statistics, machine learning, and data science consultancy and training organization.
- Specialize in solution design, technology evaluation, and prototyping.
- Contact: jmount@win-vector.com .

Win-Vector LLC

Outline

- The problem
    - Some history
- Solutions
    - Experiments/Results
- Methods
- Observations
    - Conclusions/recommendations

The Problem

The Situation

- Users (or user personas) have intrinsic unobserved ***preferences*** or valuations
  - Personas are either representatives of a group of users or a label collecting together a group of users.
  - This lets us assume we can collect a lot of per-persona data.
- We observe user ***behaviors***
  - Typical observation: we presented 5 alternatives and the user purchased one.
  - Observation contaminated both by presentation position and alternative items in the presentation.
- Can we estimate persona preferences from persona behaviors?
  - Often called “learning to rank.”
  - We can use estimated preferences to plan (a lot more than just predicting future list behaviors).

Our Goal

- Estimate the user (or user cohort) *intrinsic* preferences (independent of presentation position and other items).
  - Allows us to rate items with respect to user to retrieve, sort, estimate utilities and so on.
  - Eliminates some systems, such as **`xgb.XGBRanker`** (as it reproduces predictions over whole lists
  - From: https://xgboost.readthedocs.io/en/stable/python/examples/learning_to_rank.html#sphx-glr-python-examples-learning-to-rank-py

```
    X_test, clicks_test, y_test, qid_test = sort_ltr_samples(
        test.X,
        test.y,
        test.qid,
        test.click,
        test.pos,
    )
…
ranker.predict(X_test)
```

(notice the `qid` column, which is means data is in presentation and we `xgb.XGBRanker` is compute probability of selection with respect to the alternatives, not utility. Would have to "sum out" many proposed panels to get intrinsic utility.)

Example Problems

- User is shown 5 products online and clicks on one.
- User tastes 5 wines and buys at most one.
  - (repeat with 100 users thought to be in same persona)

We **wish** they would tell us their numerical valuation of each wine!

Our “Wine” Example

- Artificial Problem
  - Know answer
  - 3 explanatory variables: x1, x2, x3
- Visible Data
  - Panels of 5 wines tasted in one sitting
  - explanatory variables
  - selected (if any) wine
- Many repetitions over many customers thought to be in same persona cluster.

	group	position	example_index	x1	x2	x3	score	hidden_activation	selected
0	0	0	7	0.00	1.00	1.00	0.295	False	False
1	0	1	2	0.00	0.00	0.10	0.257	False	False
2	0	2	3	0.00	1.00	1.00	0.304	False	False
3	0	3	8	0.00	1.00	1.00	0.288	False	False
4	0	4	10	0.00	1.00	1.00	0.348	False	False
5	1	0	1	1.00	6.00	0.00	0.546	True	False
6	1	1	0	12.00	1.00	0.00	0.796	True	True
7	1	2	3	0.00	1.00	1.00	0.326	True	False
8	1	3	7	0.00	1.00	1.00	0.309	False	False
9	1	4	2	0.00	0.00	0.10	0.210	True	False
10	2	0	6	0.00	1.00	1.00	0.349	False	False
11	2	1	0	12.00	1.00	0.00	0.817	True	True
12	2	2	5	0.00	1.00	1.00	0.297	False	False
13	2	3	10	0.00	1.00	1.00	0.327	True	False
14	2	4	8	0.00	1.00	1.00	0.327	False	False

The (Assumed) Data Generating Process

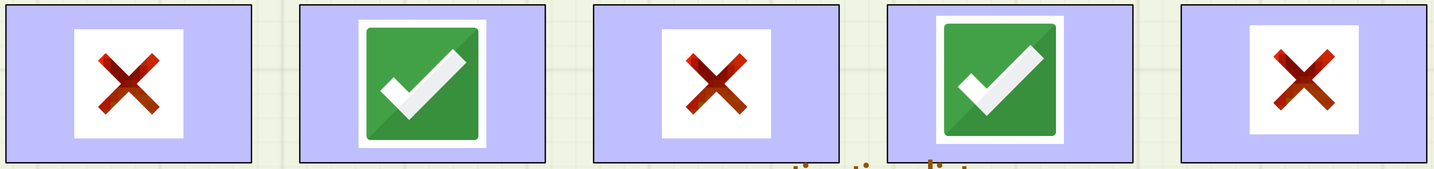
User considers 5 alternatives

(unobserved) explanatory variables converted into activation probabilities (scores)

0.349	0.817	0.297	0.327	0.327

wish we know this

(unobserved) probabilities converted into “interested” activations check-marks



activation list

(observed) checked item with highest activation score is picked

selection list

experiment design

model structure

$f_\beta(x)$ (fit for β)

presentation censoring process

Win-Vector LLC

List-Wise Observations

Outcome (y):

Encodes as:

"picked wine 4"

Simulates a single pick from a single tasting of each wine.

Leaning to Rank History

- Tradition: each field develops solutions ignoring all other fields
  - BIG topic in search engines
    - Joachims, T. (2002), "Optimizing Search Engines using Clickthrough Data", *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*
    - Tie-Yan Liu (2009), "Learning to Rank for Information Retrieval", *Foundations and Trends in Information Retrieval*, **3** (3): 225–331
- Huge survey: https://en.wikipedia.org/wiki/Learning_to_rank
  - Claims that logistic regression does not work
    - “Bill Cooper proposed logistic regression for the same purpose in 1992 [19] and used it with his Berkeley research group to train a successful ranking function for TREC. Manning et al.[40] suggest that these early works achieved limited results in their time due to little available training data and poor machine learning techniques.”
- May over sell presentation hygiene.

Solutions

Our Thesis

- Factor overall system performance into the product of:
  - "**Model Structure**" (how well the explanatory variables can reproduce scores).
    - We used logistic regression (great at converting yes/no observations back into coefficients and probability scores).
  - "**Presentation Un-Censoring Hygiene**" (how well we account for the censoring effects of presentation on the problem). We are treating this as a *required user supplied causal structure*.
    - May involve joining in some "facts about presentation" and "user profile" interactions.
- To improve: must know where we *actually* are losing quality.
  - Simulate to attribute quality loss to each component separately.

Maybe Things Can Be Easy?

- If: most of the unobserved activation lists have fewer than two activations.
- Then:
    - Most of the unobserved activation lists are then identical to the observed selection lists.
    - Problem is solvable by direct item-wise logistic regression.
- Example: online shopping with low activation rates.

Maybe Things are Difficult?

- Presentation position may be a strong influence
    - Users may be biased to pick earlier presentation positions, forget earlier positions, or even not even try/look-at later positions!
    - Item quality may correlate with presentation position.
    - What other items are in the list or panel having an effect (such as high priced irrelevant alternatives).
- Data is noisy
    - User may make different choice when re-presented
- Data is low fidelity or censored
    - Selecting 1 out of 5 positions is only 2.3 bits of information
- **No-select lists *may* be coming from an unknown mixture of two sources**
    - **Motivated Shoppers (with intent to buy, so non-buying strong evidence against all presented wines).**
    - **Idle Browsers (with no intent to buy, so non-buying against any wine).**
    - **These are not the same persona! However may be able to say who is who at individual level!**

Our Experiment: Activation Probabilities

	example_index	x1	x2	x3	data generation process	model from visible selections	model from pair differences	Stan model MCMC	Stan model optimizer
0	0	12.0	1.0	0.0	0.80	0.80	0.99	0.80	0.80
1	1	1.0	6.0	0.0	0.55	0.39	0.94	0.56	0.56
2	2	0.0	0.0	0.1	0.24	0.04	0.51	0.26	0.32
3	3	0.0	1.0	1.0	0.31	0.10	0.73	0.31	0.34

The Methods

Example Feature Encoding

$$x(item = 6425, position = 2) =$$

	6425
fixed acidity	6.0
volatile acidity	0.34
citric acid	0.29
residual sugar	6.1
chlorides	0.046
free sulfur dioxide	29.0
total sulfur dioxide	134.0
density	0.99462
pH	3.48
sulphates	0.57
alcohol	10.7
is_red	False
posn_0	0
posn_1	0
posn_2	1
posn_3	0
posn_4	0

Feature table lookup

Encoding of presentation position. Could also encode demographics of participant Especially useful if we interact the demographic variables with the feature variables.

Difference Encoding

- Exploit linear structure to directly encode differences
  - Explanatory variables: encode comparison pairs.
    - For example: item 6425 in position 2 **picked** and item 1569 in position 2 **not picked** encoded as:

$$x_i = x(item = 6425, position = 2) - x(item = 1569, position = 0)$$

    - The subtraction enforces that features have the same interpretation in picks and non-picks.
    - Outcomes encodes as “True”.
- Outcomes/dependent variables must vary
  - So also encode a “False” outcome for $-x_i$
- Encoding not natural or obvious.

Stan Model

- Guess: β (coefficient vector including intercept), ζ (per visible chosen selection vector noise term).
- Let $x_{i,j}$ denote the vector of explanatory values for the ith example's jth alternative

  We model the hidden scores as being distributed normal$(x_{i,j} \cdot \beta, \sigma)$ (or $\zeta_i \sim$ normal$(0,\sigma)$).

  For lists with no selection:

  $$\mathrm{P}[data_i \,|\, \beta, \zeta_i] = \prod_j (1 - \mathrm{sigmoid}(x_{i,j} \cdot \beta))$$

  (No selection arrises from failing to select any of the alternatives.)

  For lists with a selection s:

  $$\mathrm{P}[data_i \,|\, \beta, \zeta_i] = \mathrm{sigmoid}(x_{i,s} \cdot \beta + \zeta_i) \prod_{j, j \neq s} (1 - \mathrm{sigmoid}(x_{i,j} \cdot \beta)(1 - \Phi(x_{i,j} \cdot \beta \,|\, x_{i,s} \cdot \beta + \zeta_i, \sigma))$$

  (For a selection to prevail it must be activated, and none of the alternative must simultaneously activate and exceed the selection's score. We condition on an explicit ζ_i is to make the terms conditionally independent and justify multiplying.

- Set up Stan to sample β, ζ such that $\mathrm{P}[\beta, \zeta]\mathrm{P}[data \,|\, \beta, zeta]$ is large. These are going to be such that $\mathrm{P}[\beta \,|\, data]$ is large (i.e. the β implied by our observation data).

Our Experiment: Model Coefficients

	model	beta_0	x1	x2	x3
0	data generation process	-1.20	0.20	0.20	0.20
1	model from visible selections	-3.13	0.34	0.39	0.50
2	model from pair differences	0.00	0.32	0.41	0.57
3	Stan model MCMC	-1.05	0.19	0.18	0.05
4	Stan model optimizer	-0.73	0.16	0.13	-0.06

	example_index	x1	x2	x3	data generation process	model from visible selections	model from pair differences	Stan model MCMC	Stan model optimizer
0	0	12.0	1.0	0.0	0.80	0.80	0.99	0.80	0.80
1	1	1.0	6.0	0.0	0.55	0.39	0.94	0.56	0.56
2	2	0.0	0.0	0.1	0.24	0.04	0.51	0.26	0.32
3	3	0.0	1.0	1.0	0.31	0.10	0.73	0.31	0.34

Observations

- Model structure likely more important than the presentation hygiene.
    - Only possible to determine if you can measure.
- Stan is great for prototyping solution methods and experimenting with how much model structure and presentation hygiene you wish to capture.
    - Unfortunately method was brittle.
    - Stan running slow often a sign of bad model structure or degenerate data.
- Logistic regression may take some steps to justify, but often works well and is fast.

Tools Used

- Stan, Python, and R
  - All code and data shared here:

    https://github.com/WinVector/Examples/tree/main/rank

    generate_example.ipynb

    LearningToRank.pdf

Thank you