TBD

Ravi Kumar
Google
Mountain View, CA
Theory Behind Discrete choice

Ravi Kumar
Google
Mountain View, CA

(Joint work with Flavio Chierichetti & Andrew Tomkins)
Discrete choice

Random user

Slate

{ Slate }

Choice distribution

\{ 25\%, 10\%, 65\% \}
Discrete choice

Random user

Slate

\{ 30\%, 70\% \}

How to learn the probability distributions governing the choice in a generic slate?
Discrete choice

Random user
Discrete choice

Quickly learning the winning distributions of the slates is important for applications ... but there are exponentially many slates!
Theory of discrete choice

Universe = $[n] = \{1, 2, ..., n\}$

Slates = non-empty subsets of $[n]$

**Model.** A function $f: \text{slate} \rightarrow \text{distribution over slate}$

Discrete choice models can codify rational behavior

S and T highly overlap $\implies f(S)$ and $f(T)$ may be related
Random utility model (RUM) (Marschak 1960)

- Exists a distribution $\mathcal{D}$ on user utilities $\{[n] \rightarrow \mathbb{R}\}$

- Each user is $D \sim \mathcal{D}$ iid and will choose highest utility option in a slate $T$ (ie, $\text{argmax}_{t \in T} D(t)$)

- Highly overlapping subsets will be related
  - Eg, $\Pr[j | T] \geq \Pr[j | T \cup \{i\}]$ for $j \in T$ and $i \not\in T$

- Rational behavior $\implies$ the order of utilities determines choice
  - $\mathcal{D}$ is a distribution on permutations of $[n]$
Example

60% Random User

40% Random User

{40%, 60%} = Slate
Example

60% 

40% 

Random User

\{ 0\%, 60\%, 40\% \} = \text{Slate}
Assume a universe \([n]\) and an unknown distribution on the permutations of \([n]\).

Given a slate \(S \subseteq [n]\), let \(D_S(i)\) for \(i \in S\) be the probability that a random permutation (i.e., user) prefers \(i\) to every other element of \(S\).
Learning RUMs

**Goal.** Learn $D_S$, for all $S \subseteq [n]$
The type of queries that we allow can significantly change the hardness of the problem.

By obtaining $O((n/\varepsilon)^2)$ random independent permutations (according to the unknown distribution), one can approximate each slate’s winning distribution to within an $\ell_1$-error of $\varepsilon$.

Given a generic slate, return the winning probabilities induced by a random permutation chosen in the set of samples.
Is this reasonable?

It is easier to ask/infer the preferred option among those in a slate.

The random permutation query is infeasible in many applications.
RUM learning

• We study RUM learning from the oracle perspective

• The system can propose slates to random users and observe which options they select

• An algorithm can query (adaptively or non-adaptively) some sequence $S_1, S_2, \ldots$ of slates to obtain their (approximate) winning distributions $D_{S_1}(\cdot), D_{S_2}(\cdot), \ldots$. 
Oracles for RUMs

Given a slate $S$

- $\text{max-sample}(S)$: picks an unknown random permutation $\pi$, and returns the element of $S$ with maximum rank in $\pi$

- $\text{max-dist}(S)$: returns $D_S(i)$, for all $i \in S$, ie, the probability that $i$ wins in $S$ given a random permutation
A general lower bound

- Even with the more powerful \texttt{max-dist} oracle, $\Omega(2^n)$ queries are needed to learn $D_S$ exactly.

- With $o(2^n)$ queries, there will be some set where the expected total variation distance is going to be $\Omega(2^{-3n/2})$.

- Smaller number of queries $\implies$ more error.
What is the hope?

There are only a few types of users.
If there are only $k$ types of users, then

- Can reconstruct exactly all the $D_S$'s with $O(nk)$ calls to the $\text{max-dist}$ oracle

- Can reconstruct all the $D_S$'s to within $\ell_1$-error of $\varepsilon$ with $\text{poly}(n, k, \varepsilon)$ calls to the $\text{max-sample}$ oracle
Efficient versions of RUMs

- Few user types
- Multinomial logits (MNLs)
Multinomial logit (MNL) (Bradley & Terry 1952; Luce 1959)

- Classical special case of RUMs

**Model.** Given a universe U of items and a positive weight $a_u$ for each item $u$ in U

For a subset (slate) $S$ of U, the probability of choosing $u$ in slate $S$ is proportional to $a_u$

$$\Pr[\text{choosing } u \text{ in } S] = \frac{a_u}{\sum_{v \in S} a_v}$$
MNL example

Pick the next item in the permutation at random between the remaining ones, with probability proportional to its weight.
1-MNL learning

**Goal.** Learn the weight $a_i$ for each $i \in [n]

Assume for a slate $S$ we get the choice distribution $D_S(\cdot)$ exactly (max-dist oracle)

For $i = 1, \ldots, n-1$, query the MNL using slate $\{i, n\}$

to get the choice distribution $D_{i,n}(\cdot)$

$$
(a_i / (a_i + a_n), \quad a_n / (a_i + a_n))
$$
A linear system

\[
\frac{a_n}{(a_1 + a_n)} = D_{1,n}(n)
\]
\[
\frac{a_n}{(a_2 + a_n)} = D_{2,n}(n)
\]

\[\ldots\]
\[\sum a_i = 1\]

Solve the resulting system of linear equations to obtain the weights
1-MNL can be learnt with $O(n)$ queries and slates of size 2.
How good are 1-MNLs?

<table>
<thead>
<tr>
<th></th>
<th>~50%</th>
<th>~50%</th>
<th>~40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEGAN FOOD</td>
<td>1</td>
<td>ε</td>
<td>4</td>
</tr>
<tr>
<td>CRAFT</td>
<td>1</td>
<td>1</td>
<td>ε</td>
</tr>
<tr>
<td>Vegan</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
</tbody>
</table>
Weakness of $1$-MNLs

$1$-MNLs are insufficient to capture common settings
Mixture of MNLs

- Modeling distinct populations with 1-MNL causes the problem
- Allowing a mixture of population, with a population-specific MNL, can solve the problem
  - New items need not cannibalize equally from all other items
  - New vegan restaurant affects only vegans
2-MNL mixture: Given a universe $U$ of items and positive weights $a_u$ and $b_u$ for each item $u$ in $U$.

For a slate $S$, the probability of choosing $u$ in $S$ equals

$$\gamma \cdot \frac{a_u}{\sum_{v \in S} a_v} + (1 - \gamma) \cdot \frac{b_u}{\sum_{v \in S} b_v}$$

Uniform mixture when $\gamma = 1/2$
Power of MNL mixtures

MNL mixtures can approximate arbitrarily well any RUM (McFadden & Train 2000)
The big picture

Choice models

RUMs

k-MNLs

1-MNLs
2-MNL learning

- **Goal**: Learn weights $a_i, b_i$ for each $i \in [n]$

- Assume for a slate $S$ we get the choice distribution $D_S(\cdot)$ exactly

- Can show 2-slates are not enough to learn
2-MNL learning with 3-slates

- Query the MNL using slates \{i, j\} and \{i, j, k\} to get the choice distributions \(D_{i,j}(\cdot)\) and \(D_{i,j,k}(\cdot)\)

\[
2 D_{i,j}(i) = \frac{a_i}{a_i + a_j} + \frac{b_i}{b_i + b_j}
\]

\[
2 D_{i,j,k}(i) = \frac{a_i}{a_i + a_j + a_k} + \frac{b_i}{b_i + b_j + b_k}
\]
A polynomial system

\[ 2 D_{i,j}(i) = \frac{a_i}{a_i + a_j} + \frac{b_i}{b_i + b_j} \]

\[ 2 D_{i,k}(i) = \frac{a_i}{a_i + a_k} + \frac{b_i}{b_i + b_k} \]

\[ 2 D_{j,k}(j) = \frac{a_j}{a_j + a_k} + \frac{b_j}{b_j + b_k} \]

\[ 2 D_{i,j,k}(i) = \frac{a_i}{a_i + a_j + a_k} + \frac{b_i}{b_i + b_j + b_k} \]

\[ 2 D_{i,j,k}(j) = \frac{a_j}{a_i + a_j + a_k} + \frac{b_j}{b_i + b_j + b_k} \]

\[ a_i + a_j + a_k = 1, \quad b_i + b_j + b_k = 1 \]
Theorem. For any uniform 2-MNL and for any set of 3 elements $S = \{i, j, k\}$, the choice distributions of all the subsets of $S$ determine uniquely the weights of $i, j, k$ in each of the two MNLs.

Proof steps.

- Partition the solution space in a discrete number of regions.
- Show that at most one region can contain feasible solutions and give combinatorial algorithm to determine it.
- Use the structure of the generic region to prove uniqueness.
Patching the unique solutions

- Query slates \{1, 2, 3\}, \{1, 4, 5\}, \{1, 6, 7\}, ..., 

- Find \(s, t \in [n]\) such that \(a_s/a_t \neq b_s/b_t\)
  
  - If \(a_i/a_j = b_i/b_j\) for all \(i, j\), it is a 1-MNL

- Query slates \{1, s, t\}, \{2, s, t\}, \{3, s, t\}, ...

- \(a_i = a_{1,s,t}(i) \cdot a_s / a_{1,s,t}(s); b_i = b_{1,s,t}(i) \cdot b_s / b_{1,s,t}(s)\)
2-MNLs: Main results

**Theorem.** There is an adaptive algorithm performing max-dist queries on $O(n)$ slates of sizes 2 and 3, that reconstructs the weights of any uniform 2-MNL system on $n$ elements.

**Theorem.** There is a non-adaptive algorithm performing max-dist queries on $O(n^2)$ slates of sizes 2 and 3, that reconstructs the weights of any uniform 2-MNL system on $n$ elements.
Conclusions

• We studied a number of algorithmic problems related to discrete choice

• We believe this class of problems is theoretically important and relevant in practice
Some open questions

- What is the relative power of the $\text{max-sample} / \text{max-dist}$ oracles?
- How well can one approximate general mixtures of MNLs with the two oracles?
- Identifiability of non-uniform 2-MNLs, $k$-MNLs
- Distribution testing questions
Thank you!

Questions/Comments
ravi.k53 @ gmail