Consumer choices are driven by their preferences. Therefore I attempt to infer a consumer’s utility function by observing the choices they make.

The input to my algorithm is a set of products with known features. I use softmax regression and neural networks to output a vector of probabilities which is then used to predict the top rank item in the set of products. I compare the performance of both models across the number of training examples, and two simulated utility functions.

### Data

The data in this project is simulated where the consumer has a utility $U$ for each product $r$:

$$U(r) = \theta^T z_r + g(z_i) - \rho_i + \epsilon_{i,r}$$

I test across two utility simulations: **Linear** where $g(z_i) = 0$, and **Non-Linear** where $g(z_i)$ is sum of indicator functions.

$$g(z_i) = \sum_{j=1}^{k} 50 \cdot [z_{ij} > 50]$$

**Notation:**
- $N$: number of rounds, $r \in 1, \cdots, N$, with $m$ products in each round
- $k$: features for each product, $z_{ij}$ is product $i$’s features
- $Z_r \in \mathbb{R}^{m \times k}$: the features of the products in round $r$
- $c_r \in \mathbb{R}^m$: the one hot vector of the top ranking product $r$
- $S_r$: set of products available in a round $r$
- $\rho_i$: price of product $i$
- $w^{(i)} \in \mathbb{R}^m$: the predicted score or utility vector

### Linear Utility

Softmax regression performs well in low data regimes. If the model and features are well specified, the linear model outperforms a neural network and achieves near optimal performance.

### Non-Linear Utility

The neural network finds non-linear trends even in moderate size data regimes. Softmax regression fails when trying to fit a non-linear function.

### Top 1 Recall

<table>
<thead>
<tr>
<th>$N$</th>
<th>Oracle Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.9145</td>
<td>0.8333</td>
</tr>
<tr>
<td>1000</td>
<td>0.6181</td>
<td>0.5266</td>
</tr>
</tbody>
</table>

### Mean Reciprocal Rank

<table>
<thead>
<tr>
<th>$N$</th>
<th>Oracle Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.9145</td>
<td>0.7800</td>
</tr>
<tr>
<td>1000</td>
<td>0.6139</td>
<td>0.7668</td>
</tr>
</tbody>
</table>

**Discussion**

As expected, the linear softmax regression performs well in low data regimes and when faced with a linear model. With $N = 1000$ observations it recovers the linear utility function completely and predicts as well as the Oracle model (which knows the utility scores).

The neural network suffers from over fitting in low data regimes, and does not match the softmax regression in the linear case even with more data. However, the neural network model fits the nonlinear utility function well and outperformed the softmax regression at $N = 100$ observations.

**Future Work:** The next steps are to apply these models to real data, which requires more complex methods like latent classes to group customers together.

### References