NetRec: Product Recommendation

dmcquil2@illinois.edu & mcconne7@illinois.edu
Overview

- Picking a good recommendation strategy is extremely important because a bad strategy can cause users to lose faith in application.
- Bad recommendations make it impossible to deliver a nice personal experience to the end user.
- NetRec: a mechanism to display relevant and useful products to a consumer based off their preferences.
NetClus

- An intermediate step in our process algorithm
- Works by using the EM algorithm to maximize the likelihood of model parameters
NetClus: Steps

1. Partition the target nodes into different clusters.
2. Build the ranking based generative model:
   a. \( \{ P(x|C_{kt}) \}_{k=1}^{K} \)
3. Calculate the posterior probabilities for the target objects
4. Calculate mean attributes of each cluster:
   a. \( \{ p(C_{kt}|x) \} \{ 1/||Xk|| \sum_{x \in X} p(C_{kj}|x) \}_{k=1}^{K} \}
5. Adjust clusters from the updated attributes
6. Repeat until convergence:
   a. \( \{ C^* \}_{k=1}^{K} = \{ C \}_{k=1}^{I} = \{ C^{I-1} \}_{k=1}^{K} \)
Cluster Centers

- How do we represent cluster attributes?
  - Mean probability vector of target objects in network
  - \( d = <p(k_1|d), \ldots, p(k_K|d)> \)
- Readjust target node clusters based minimum distance using cosine similarity to cluster centers
NetClus: Results

<table>
<thead>
<tr>
<th>NetClus Products</th>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top 3</strong></td>
<td>Tri-State Corn on the Cob: 0.0065</td>
<td>Cormorant Scented Toilet Tissue: 0.0042</td>
</tr>
<tr>
<td></td>
<td>High Top Cron on the Cob: 0.0063</td>
<td>Red Wing Soft Napkins: 0.0036</td>
</tr>
<tr>
<td></td>
<td>Hermanos Garlic: 0.0060</td>
<td>Booker Low Fat Cottage Cheese: 0.0036</td>
</tr>
<tr>
<td></td>
<td>Bravo Turkey Noodle Soup: 0.0003</td>
<td>Hermanos Garlic: 0.0000</td>
</tr>
<tr>
<td></td>
<td>Blue Label Rice Soup: 0.0003</td>
<td>High Top Asparagus: 0.0000</td>
</tr>
<tr>
<td></td>
<td>Better Noodle Soup: 0.0000</td>
<td>Ebony Beets: 0.0000</td>
</tr>
</tbody>
</table>

Table 2: Results of NetClus for clusters 1 and 2

<table>
<thead>
<tr>
<th>NetClus Products</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top 3</strong></td>
<td>Better Vegetable Soup: 0.0125</td>
<td>Better Noodle Soup: 0.0030</td>
</tr>
<tr>
<td></td>
<td>Blue Label Chicken Ramen Soup: 0.0109</td>
<td>Super Chunky Peanut Butter: 0.0028</td>
</tr>
<tr>
<td></td>
<td>Pleasant Turkey Noodle Soup: 0.0105</td>
<td>Great Rye Bread: 0.0028</td>
</tr>
<tr>
<td></td>
<td>Hermanos Broccoli: 0.0000</td>
<td>American Foot-Long Hot Dogs: 0.0012</td>
</tr>
<tr>
<td></td>
<td>Fast Golden Raisins: 0.0000</td>
<td>Red Wing Scented Toilet Tissue: 0.0012</td>
</tr>
<tr>
<td></td>
<td>Tell Tale Mandarin Oranges: 0.0000</td>
<td>Walrus Light Beer: 0.0012</td>
</tr>
</tbody>
</table>

Table 3: Results of NetClus for clusters 3 and 4
NetClus: Runtime Performance

- Scales linearly
- $O(c_1|E| + c_2|N|)$
- Unfortunately, we experienced high variance in our execution times.
  - We suspect this is due to IO and other parts of our system with high constant execution time complexity.
NetRec: Diffusion Matrix

- Captures how events propagate through a network
- Usages
  - The pattern of how a virus could move across a network
  - Viral marketing message amongst a network of people
  - How fast a disease could spread
  - How the purchases of products by a user influence their future purchases
NetRec: Diffusion matrix (Calculation)

- Calculated in terms of PathSim and an indicator matrix to contain user interest

- \[ R_{u_i,e_j}^{(\ell)} = s(u_i,e_j|P^{(\ell)}) = \sum_{e \in I} \frac{2 \cdot R_{u_i,e} \cdot M_{ee_j}}{M_{ee} + M_{e_j e_j}} \]

- \[ R_{u_i,e} = \begin{cases} 
1 & \text{if interaction is observed} \\
0 & \text{otherwise} 
\end{cases} \]

\[ M_{ee} = \left\{ p_{i \rightarrow j} : p_{i \rightarrow j} \in P \right\} \]

I = The set of all items

\( e_j = \) a product, j

\( u_i = \) a user, i
NetRec: Recommendation Model

- Utilizes NetClus to determine user-cluster similarity
  - \( \text{netsim}(C_k, u_i) = p(C_k | u_i) \)

- Determines recommendation score using a latent variable \( \Theta \) to represent the weights for a given meta-path
  - \( r^*(u_i, e_j) = \sum_{k=1}^{c} \text{netsim}(C_k, u_i) \sum_{\ell=1}^{L} \theta_{\ell} \cdot R_{ij}(\ell) \)
NetRec: Parameter Estimation

- $\Theta$ estimated through the use of SGD with an objective function
- Objective function
  \[
  O = -\ln p(\Theta|R) = -\ln p(R|\Theta)p(\Theta) = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \ln p(e_a > e_b; u_i|\Theta) + \lambda\|\Theta\|_2^2
  \]
  \[
  = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \ln \sigma(r(u_i, e_a) - r(u_i, e_b)) + \lambda\|\Theta\|_2^2
  \]
- Gradient
  \[
  \frac{\delta O}{\delta \theta} = -\sum_{u_i \in U} \sum_{(e_a > e_b) \in R_i} \frac{e^{-r_{i,ab}}}{1 + e^{-r_{i,ab}}} \frac{\delta}{\delta \theta} r_{i,ab} + \lambda \theta
  \]
  where:
  \[
  r_{i,ab} = r(u_i, e_a) - r(u_i, e_b)
  \]
- Update Rule
  \[
  \theta = \theta - \alpha \frac{1}{|U|} \frac{\delta O}{\delta \theta}
  \]
NetRec: Experiment - Data

- Utilized supermarket purchases from 1998
  - January - November (training data)
  - December (test data)

- Data Schema
  - Product_id - unique id for a given product
  - Time_id - time the product was purchased
  - Customer_id - unique id for a customer
  - Brand_name - brand name for a product
  - Product_name - description name for a product
  - Product_classid - roll-up id for subcategory and category pair
  - Product_subcategory - leaf category name for a product
  - Product_category - roll-up category name for a product
NetRec: Experiment - Information Networks

Purchase information network

User Information Network (Star Schema)

Network utilized to perform meta-path based user-product ranking

Network utilized for calculating user-cluster similarity using NetClus
NetRec: Experiment - Results

Nodes: 2787, Edges: 27855

Accuracy Measure

\[
C_{u_i} = \sum_{e \in T_{u_i}} I\left( r(u_i, e) > \frac{1}{|T_{u_i}|} \sum_{e' \in T_{u_i}} r(u_i, e') \right)
\]

\[
C = \sum_{u_i \in U} F = \left[ \sum_{u_i \in U} \sum_{e \in T_{u_i}} 1 \right] - C
\]

\[
\text{Accuracy} = \frac{C}{C + F}
\]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetRec</td>
<td>99.23</td>
<td>77.5</td>
</tr>
<tr>
<td>NetRec-NMF</td>
<td>88.55</td>
<td>77.4</td>
</tr>
<tr>
<td>HeteRec</td>
<td>97.4</td>
<td>75.5</td>
</tr>
</tbody>
</table>
Conclusion

- NetRec performed better in both precision and recall measures than NetRec-NMF and HeteRec.
- Loss is minimized by utilizing the full diffusion matrix instead of the NMF factorized alternatives.
- Loss is also minimized by utilizing a heterogeneous information network when performing user clustering.