Studies

• Constructing and Exploring Composite Items [Roy et al. 2010]
• Breaking out of the Box of Recommendations: From Items to Packages [Xie et al. 2010]
• First Approach
  • Valid and maximal packages
  • Summarization
  • Visual effect optimization

• Second Approach
  • Instance optimal algorithm
    • Top-1 composite algorithm
    • Top-k composite algorithm
  • Greedy Algorithm
First Approach

- Valid and maximal packages
  - Budget
  - Largest compatible set of satellite items

- Summarization
  - Using Greedy and randomized summarization algorithm
  - Reduce volume

- Visual effect optimization
  - Using visual effect optimization NP and heuristic visual effect optimization
  - No overlapping
  - Order
Second Approach

- Composite system
- Access to more informative sources
  - eg. databases, websites.
- Multiple recommender system
Second Approach

• Instance optimal algorithm
  • Top-1 composite algorithm
    • Maximum value
    • Optimal value
    • Reduction of size of the packages
  • Top-k composite algorithm
    • k is a small constant

• Greedy Algorithm
First approach

Drawbacks
Maximal Packages
Summarization Drawback

• 255 packages can be constructed from the 8-item package $p_1$ and 31 packages can be constructed from the 5-item package $p_3$. Formula: $2^n - 1$

• Each $M_i$ (set of packages) may require the summation of an exponential number of terms. As a result, summarization by maximizing coverage turns out to be a hard problem.
Visual Effect Optimization drawback

• Condition:
  • Graph can transform into a set of packages in $S$ in polynomial time.
  • Completed graph
  • Hamilton path must exist
Hamilton path and complete graph

- Hamilton Path: It is a graph path between two vertices of a graph that visits each vertex exactly once.
- Complete Graph: It is a graph in which each pair of graph vertices is connected by an edge.
Heuristic Visual Effect Optimization

Drawback

• The algorithm is not guaranteed to find the optimal ordering. Sometimes, the algorithm will fail to find one of the two optimal orderings. For examples,

- $p_1 = (s_1\text{case}, s_1\text{charger}, s_1\text{kit}, s_1\text{cable}, s_1\text{speaker}, s_2\text{screen}, s_1\text{pen})$,
- $p_2 = (s_1\text{case}, s_1\text{charger}, s_3\text{cable}, s_1\text{screen}, s_1\text{pen})$,
- $p_3 = (s_1\text{case}, s_4\text{charger}, s_2\text{cable}, s_3\text{speaker}, s_1\text{screen}, s_1\text{pen})$,
- $p_4 = (s_2\text{case}, s_4\text{charger}, s_2\text{cable}, s_3\text{speaker}, s_1\text{screen}, s_1\text{pen})$
There is two group $G_1 \text{case} = \{p_1, p_2, p_3\}$ and $G_2 \text{case} = \{p_4\}$. It is important to deterministically select the next package such that its addition incurs the least penalty with respect to the previously added package. Otherwise, a random selection between $p_2$ and $p_3$ in the third step may generate an ordering such as $(p_1, p_4, p_3, p_2)$, which is worse than the ordering that this algorithm produces.
GreedySummary Set and MaxCompositeItemSet algorithms produce the full set of maximal packages. The process of generating the full set of maximal packages alone is quite time-consuming.
Summarization Algorithms Performance

Performance of Summarization Algorithms

- MaxCompositeItemSet
- GreedySummarySet
- FastGreedySummarySet
- ProbSummarySet

# Average Time (milliseconds in log scale)

Number of Representatives

Time in milliseconds:
- 5
- 10
- 15
- 20
- 25
Second approach
Experiments

• The goal of the experiments were:
  
  • Evaluate the relative quality of Inst-Opt-CR and Greedy-CR compared to the optimal algorithm, in terms of both the total and average values of the top-k packages returned.
  
  • Evaluate the relative efficiency of the algorithms with respect to the number of items accessed and the actual run time.
The table shows the quality of the top-5 composite recommendations returned by the optimal and approximation algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1st Package</th>
<th>2nd Package</th>
<th>3rd Package</th>
<th>4th Package</th>
<th>5th Package</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUM</td>
<td>AVG</td>
<td>SUM</td>
<td>AVG</td>
<td>SUM</td>
</tr>
<tr>
<td><strong>MovieLens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal</td>
<td>427</td>
<td>46.7</td>
<td>428</td>
<td>46.6</td>
<td>425</td>
</tr>
<tr>
<td>InsOpt-CR-Topk</td>
<td>386</td>
<td>47.5</td>
<td>385</td>
<td>47.4</td>
<td>-385</td>
</tr>
<tr>
<td>Greedy-CR-Topk</td>
<td>384</td>
<td>47</td>
<td>381</td>
<td>47</td>
<td>-380</td>
</tr>
<tr>
<td><strong>TripAdvisor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal</td>
<td>300</td>
<td>50</td>
<td>300</td>
<td>50</td>
<td>-300</td>
</tr>
<tr>
<td>InsOpt-CR-Topk</td>
<td>185</td>
<td>50</td>
<td>175</td>
<td>50</td>
<td>165</td>
</tr>
<tr>
<td>Greedy-CR-Topk</td>
<td>220</td>
<td>50</td>
<td>210</td>
<td>50</td>
<td>210</td>
</tr>
<tr>
<td><strong>Uncorrelated Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal</td>
<td>1092</td>
<td>36.4</td>
<td>1091</td>
<td>36.4</td>
<td>1090</td>
</tr>
<tr>
<td>InsOpt-CR-Topk</td>
<td>929</td>
<td>43.6</td>
<td>928</td>
<td>43.6</td>
<td>925</td>
</tr>
<tr>
<td>Greedy-CR-Topk</td>
<td>945</td>
<td>42.9</td>
<td>939</td>
<td>42.8</td>
<td>936</td>
</tr>
<tr>
<td><strong>Correlated Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal</td>
<td>122</td>
<td>5.3</td>
<td>122</td>
<td>5.2</td>
<td>122</td>
</tr>
<tr>
<td>InsOpt-CR-Topk</td>
<td>110</td>
<td>6.7</td>
<td>110</td>
<td>6.7</td>
<td>110</td>
</tr>
<tr>
<td>Greedy-CR-Topk</td>
<td>110</td>
<td>6.6</td>
<td>110</td>
<td>6.6</td>
<td>109</td>
</tr>
</tbody>
</table>
Results: Optimal & Greedy Algorithm

• It can be verified from the table that the approximation algorithms do indeed return top-k composite packages whose value is guaranteed to be a 2-approximation of the optimal algorithm.

• The proposed approximation algorithms often recommend packages with high average value, whereas the optimal algorithm often tries to fill the package with small cost and small value items.
Results study

- To measure the quality of the top-$k$ composite packages returned by the approximation algorithms against the optimal algorithm, a modified Normalized Discounted Cumulative Gain (NDCG) is used.

- It is clear that, while having a substantial run time advantage, the greedy algorithm can achieve a very similar overall top-$k$ package quality compared to the instance optimal algorithm.

- We also note that both approximation algorithms have a very small NDCG score.
Results study

• The running times of our algorithms on the 4 datasets
  
  • For MovieLens, TripAdvisor and the uncorrelated dataset, it can be seen that on average the greedy algorithm Greedy-CR-Topk has excellent performance in terms of both running time and access cost.

  • The only dataset where both the greedy and instance optimal algorithms have a high access cost is the correlated dataset (but notice that the greedy algorithm still has good running time).