Joint Caching and Recommendation? Get Real!
Marina Costantini, Theodoros Giannakas, Thrasyvoulos Spyropoulos
{first.last}@eurecom.fr

1. Bridging Caching and Recommendations!
   - Fact #1: Jointly designing caching and recommendations can benefit both the users and the operators. [1, 2, 3]
   - Fact #2: We want to cache content that would be good to recommend for many other contents.

An important abstraction (U-graph): Contents are vertices, weighted edges reveal level of relevance between them (\(u_{ij}\)).

- Question #1: What is more important: content popularities or content relationships (captured in the "U-graph")?
- Question #2: Is it possible to “guess” a near optimal strategy when we know some key graph statistics?
- Question #3: Are there content types for which the joint optimization is more powerful than others?

Key message: Answers depend on properties of U-graph.

2. Graph Properties & Intuition
   a) Degree skewness, cache size and recommendations
      - RC and SCH perform significantly better when the degree distribution is skewed (solve a cover problem over graph \(U\)).
      - Higher cache size or fewer recommendations can confuse SCH heuristic (mistakenly caches many nearby contents)

   b) Community structure
      - benefits both algorithms that take graph structure into account.

   Figure 1 - Performance in graphs with different degree distribution for different cache sizes (C) and number of recommendations (N).

   Figure 2 - Different number of clusters (n) and rewiring probability (p).

3. Visualization of the strategies
   - Pop does not consider the graph structure or the soft hits.
   - SCH chooses nodes with many shared connections.
   - RC finds a good compromise between popularity and cover.

4. Performance on Real Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of nodes (K)</th>
<th>Mean degree</th>
<th>Degree skewness</th>
<th>Popularity skewness</th>
<th>Clustering coefficient</th>
<th>Number of clusters (n)</th>
<th>nK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon VG</td>
<td>5614</td>
<td>22</td>
<td>0.0029</td>
<td>3.11</td>
<td>11.6</td>
<td>0.28</td>
<td>14</td>
</tr>
<tr>
<td>Amazon Apps</td>
<td>8229</td>
<td>15.9</td>
<td>0.0019</td>
<td>11.94</td>
<td>12.5</td>
<td>0.08</td>
<td>15</td>
</tr>
<tr>
<td>Amazon TV</td>
<td>2789</td>
<td>7.7</td>
<td>0.0027</td>
<td>2.97</td>
<td>10.9</td>
<td>0.21</td>
<td>26</td>
</tr>
<tr>
<td>Last FM</td>
<td>3506</td>
<td>4.2</td>
<td>0.0012</td>
<td>3.60</td>
<td>10.1</td>
<td>0.13</td>
<td>138</td>
</tr>
<tr>
<td>MovieLens</td>
<td>3306</td>
<td>25.4</td>
<td>0.0077</td>
<td>2.41</td>
<td>2.07</td>
<td>0.54</td>
<td>25</td>
</tr>
<tr>
<td>YouTube</td>
<td>2098</td>
<td>5.3</td>
<td>0.0025</td>
<td>1.41</td>
<td>7.57</td>
<td>0.37</td>
<td>38</td>
</tr>
</tbody>
</table>

Table 2 - Trace parameters. Skewness was measured Pearson’s coefficient.

References
1. Pavlos Sermpezis et al. "Soft Cache Hits: Improving Performance through Recommendation and Delivery of Related Content", in IEEE Journal on Selected Areas in Communications (JSAC), 2018